Notes de mise à jour d'Aurora MySQL

# Amazon Aurora

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

# Amazon Aurora: Notes de mise à jour d’Aurora MySQL

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

Les marques et la présentation commerciale d’Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n’est pas d’Amazon, d’une manière susceptible de créer une confusion parmi les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d’Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

# Table of Contents

Notes de mise à jour d'Aurora MySQL ..... 1
Calendriers de publication d'Aurora MySQL ..... 2
Calendrier des versions majeures d'Aurora MySQL ..... 2
Calendrier des versions mineures d'Aurora MySQL ..... 3
Aurora MySQL version 3 ..... 6
Mises à jour d'Aurora MySQL : 2013-11-2020 (version 3.11.0, compatible avec MySQL 8.0.43) ..... 7
Améliorations ..... 8
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 13
Mises à jour d'Aurora MySQL : 2025-11-20 (version 3.10.2, compatible avec MySQL 8.0.42) ..... 13
Améliorations ..... 14
Mises à jour d'Aurora MySQL : 2025-09-30 (version 3.10.1, compatible avec MySQL 8.0.42) ..... 15
Améliorations ..... 16
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 18
Mises à jour d'Aurora MySQL : 31/07/2020 (version 3.10.0, compatible avec MySQL 8.0.42) ..... 18
Nouvelles fonctionnalités ..... 19
Améliorations ..... 20
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 22
Mises à jour d'Aurora MySQL : 14/05/2020 (version 3.09.0, compatible avec MySQL 8.0.40) ..... 23
Points forts de la publication ..... 23
Améliorations ..... 24
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 27
Mises à jour d'Aurora MySQL : 2025-04-07 (version 3.08.2, compatible avec MySQL 8.0.39) ..... 28
Améliorations ..... 29
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 30
Mises à jour d'Aurora MySQL : 17/02/2020 (version 3.08.1, compatible avec MySQL 8.0.39) ..... 30
Améliorations ..... 29
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 30
Mises à jour d'Aurora MySQL : 2024-11-18 (version 3.08.0, compatible avec MySQL 8.0.39) ..... 31
Nouvelles fonctionnalités ..... 32
Améliorations ..... 34
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 40
Mises à jour d'Aurora MySQL : 2024-07-23 (version 3.07.1, compatible avec MySQL 8.0.36) ..... 40
Améliorations ..... 41

Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 42
Mises à jour d'Aurora MySQL : 04/06/2022 (version 3.07.0, compatible avec MySQL 8.0.36) ..... 42
Améliorations ..... 42
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 46
Mises à jour d'Aurora MySQL : 26/06/2024 (version 3.06.1, compatible avec MySQL 8.0.34) ..... 47
Améliorations ..... 48
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 51
Mises à jour d'Aurora MySQL : 07/03/2021 (version 3.06.0, compatible avec MySQL 8.0.34) ..... 51
Nouvelles fonctionnalités ..... 52
Améliorations ..... 48
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 51
Mises à jour d'Aurora MySQL : 2024-01-31 (version 3.05.2, compatible avec MySQL 8.0.32) ..... 57
Améliorations ..... 57
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 59
Mises à jour d'Aurora MySQL : 21/11/2023 (version 3.05.1, compatible avec MySQL 8.0.32) ..... 59
Améliorations ..... 60
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 60
Mises à jour d'Aurora MySQL : 30/10/2023 (version 3.05.0.1, compatible avec MySQL 8.0.32), Bêta ..... 61
Améliorations ..... 61
Mises à jour d'Aurora MySQL : 25/10/2023 (version 3.05.0, compatible avec MySQL 8.0.32) ..... 62
Améliorations ..... 62
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 67
Mises à jour d'Aurora MySQL : 05/05/2020 (version 3.04.4, compatible avec MySQL 8.0.28) ..... 67
Améliorations ..... 68
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 71
Mises à jour d'Aurora MySQL : 26/06/2024 (version 3.04.3, compatible avec MySQL 8.0.28) ..... 71
Améliorations ..... 72
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 74
Mises à jour d'Aurora MySQL : 15/03/2020 (version 3.04.2, compatible avec MySQL 8.0.28) ..... 74
Améliorations ..... 75
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 78
Mises à jour d'Aurora MySQL : 13/11/2023 (version 3.04.1, compatible avec MySQL 8.0.28) ..... 78
Améliorations ..... 80
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 81
Mises à jour d'Aurora MySQL : 31/07/2023 (version 3.04.0, compatible avec MySQL 8.0.28) ..... 82

Améliorations ..... 83
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 89
Mises à jour d'Aurora MySQL : 2023-12-08 (version 3.03.3,) (obsolète) ..... 90
Améliorations ..... 91
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 92
Mises à jour d'Aurora MySQL : 29/08/2023 (version 3.03.2) (obsolète) ..... 92
Améliorations ..... 93
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 95
Mises à jour d'Aurora MySQL : 11 mai 2023 (version 3.03.1) (obsolète) ..... 95
Améliorations ..... 96
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 98
Mises à jour d'Aurora MySQL : %01 (version 3.03.0) (obsolète) ..... 98
Améliorations ..... 99
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 102
Mises à jour d'Aurora MySQL : 17/04/2023 (version 3.02.3) (obsolète) ..... 103
Améliorations ..... 104
Mises à jour d'Aurora MySQL : 03.11-18 (version 3.02.2) (obsolète) ..... 105
Améliorations ..... 106
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 102
Mises à jour d'Aurora MySQL : 09.09-07 (version 3.02.1) (obsolète) ..... 109
Améliorations ..... 109
Mises à jour d'Aurora MySQL : 20/04/2018 (version 3.02.0) (obsolète) ..... 111
Améliorations ..... 112
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 114
Mises à jour d'Aurora MySQL : 04/04/15 (version 3.01.1) (obsolète) ..... 114
Améliorations ..... 115
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 117
Mises à jour d'Aurora MySQL : 2021-11-18 (version 3.01.0) (obsolète) ..... 117
Améliorations ..... 118
Aurora MySQL version 2 ..... 119
Mises à jour d'Aurora MySQL : 2025-04-09 (version 2.12.5, compatible avec MySQL 5.7.44) - Version RDS Extended Support ..... 123
Améliorations ..... 123
Mises à jour d'Aurora MySQL : 2024-10-23 (version 2.12.4, compatible avec MySQL 5.7.44) - Version RDS Extended Support ..... 124
Améliorations ..... 123

Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 125
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 126
Compatibilité avec MySQL 5.7 ..... 126
Mises à jour d'Aurora MySQL : 2024-07-09 (version 2.12.3, compatible avec MySQL 5.7.44) - Version RDS Extended Support ..... 126
Améliorations ..... 127
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 130
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 130
Compatibilité avec MySQL 5.7 ..... 130
Mises à jour d'Aurora MySQL : 19/03/2020 (version 2.12.2, compatible avec MySQL 5.7.44) - Version RDS Extended Support ..... 131
Améliorations ..... 132
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 133
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 133
Compatibilité avec MySQL 5.7 ..... 134
Mises à jour d'Aurora MySQL : 28/12/2023 (version 2.12.1, compatible avec MySQL 5.7.40) - Version RDS Extended Support ..... 134
Améliorations ..... 135
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 138
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 138
Compatibilité avec MySQL 5.7 ..... 138
Mises à jour d'Aurora MySQL : 25/10/2023 (version 2.12.0.1, compatible avec MySQL 5.7.40) - Version RDS Extended Support (bêta) ..... 139
Améliorations ..... 140
Mises à jour d'Aurora MySQL : 25/07/2023 (version 2.12.0, compatible avec MySQL 5.7.40) - Version RDS Extended Support ..... 140
Améliorations ..... 141
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 144
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 144
Compatibilité avec MySQL 5.7 ..... 145
Mises à jour d'Aurora MySQL : 19/07/2022 (version 2.11.6, compatible avec MySQL 5.7.12) - Version RDS Extended Support ..... 145
Améliorations ..... 146
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 148
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 148
Compatibilité avec MySQL 5.7 ..... 149

Mises à jour d'Aurora MySQL : 26/03/2021 (version 2.11.5, compatible avec MySQL 5.7.12) - Version RDS Extended Support (par défaut) ..... 149
Améliorations ..... 151
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 152
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 152
Compatibilité avec MySQL 5.7 ..... 152
Mises à jour d'Aurora MySQL : 17/10/2023 (version 2.11.4, compatible avec MySQL 5.7.12) - Version RDS Extended Support ..... 153
Améliorations ..... 154
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 157
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 157
Compatibilité avec MySQL 5.7 ..... 157
Mises à jour d'Aurora MySQL : 09/06/2023 (version 2.11.3, compatible avec MySQL 5.7.12) - Version RDS Extended Support ..... 158
Améliorations ..... 159
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 160
Compatibilité avec MySQL 5.7 ..... 160
Mises à jour d'Aurora MySQL : 24/24 (version 2.11.2, compatible avec MySQL 5.7.12) - Version RDS Extended Support ..... 161
Améliorations ..... 162
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 163
Compatibilité avec MySQL 5.7 ..... 163
Mises à jour d'Aurora MySQL : 14/02/2023 (version 2.11.1, compatible avec MySQL 5.7.12) - Version RDS Extended Support ..... 164
Améliorations ..... 165
Comparaison avec Aurora MySQL Version 1 ..... 166
Compatibilité avec MySQL 5.7 ..... 167
Mises à jour d'Aurora MySQL : 25/10 (version 2.11.0, compatible avec MySQL 5.7.12) - Version RDS Extended Support ..... 167
Améliorations ..... 168
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 173
Comparaison avec Aurora MySQL Version 1 ..... 174
Compatibilité avec MySQL 5.7 ..... 175
Mises à jour d'Aurora MySQL : 01/11/2021 (version 2.10.3) (obsolète) ..... 175
Améliorations ..... 176
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 177

Comparaison avec Aurora MySQL Version 1 ..... 178
Compatibilité avec MySQL 5.7 ..... 179
Mises à jour d'Aurora MySQL : 21/01/2022 (version 2.10.2) (obsolète) ..... 179
Améliorations ..... 180
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 102
Comparaison avec Aurora MySQL Version 1 ..... 184
Compatibilité avec MySQL 5.7 ..... 184
Mises à jour d'Aurora MySQL : 21/10/2021 (version 2.10.1) (obsolète) ..... 185
Améliorations ..... 186
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 187
Comparaison avec Aurora MySQL Version 1 ..... 187
Compatibilité avec MySQL 5.7 ..... 188
Mises à jour d'Aurora MySQL : 25/05/2021 (version 2.10.0) (obsolète) ..... 188
Améliorations ..... 189
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 192
Comparaison avec Aurora MySQL Version 1 ..... 196
Compatibilité avec MySQL 5.7 ..... 196
Mises à jour d'Aurora MySQL : 12/11/2022 (version 2.09.3) (obsolète) ..... 197
Améliorations ..... 198
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 201
Comparaison avec Aurora MySQL Version 1 ..... 201
Compatibilité avec MySQL 5.7 ..... 202
Mises à jour d'Aurora MySQL : 26/02/2021 (version 2.09.2) (obsolète) ..... 202
Améliorations ..... 203
Comparaison avec Aurora MySQL Version 1 ..... 204
Compatibilité avec MySQL 5.7 ..... 205
Mises à jour d'Aurora MySQL : 11/12/2020 (version 2.09.1) (obsolète) ..... 206
Améliorations ..... 206
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 208
Comparaison avec Aurora MySQL Version 1 ..... 208
Compatibilité avec MySQL 5.7 ..... 209
Mises à jour d'Aurora MySQL : 17/09/2020 (version 2.09.0) (obsolète) ..... 209
Améliorations ..... 210
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 215
Comparaison avec Aurora MySQL Version 1 ..... 217
Compatibilité avec MySQL 5.7 ..... 217

Mises à jour d'Aurora MySQL : 06/01/2022 (version 2.08.4) (obsolète) ..... 218
Améliorations ..... 219
Comparaison avec Aurora MySQL Version 1 ..... 219
Compatibilité avec MySQL 5.7 ..... 220
Mises à jour d'Aurora MySQL : 12/11/2022 (version 2.08.3) (obsolète) ..... 221
Améliorations ..... 221
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 222
Comparaison avec Aurora MySQL Version 1 ..... 222
Compatibilité avec MySQL 5.7 ..... 223
Mises à jour d'Aurora MySQL : 28/08/2020 (version 2.08.2) (obsolète) ..... 224
Améliorations ..... 224
Comparaison avec Aurora MySQL Version 1 ..... 225
Compatibilité avec MySQL 5.7 ..... 225
Mises à jour d'Aurora MySQL : 18/06/2020 (version 2.08.1) (obsolète) ..... 226
Améliorations ..... 227
Comparaison avec Aurora MySQL Version 1 ..... 227
Compatibilité avec MySQL 5.7 ..... 228
Mises à jour d'Aurora MySQL : 06/02/2020 (version 2.08.0) (obsolète) ..... 229
Améliorations ..... 229
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 232
Comparaison avec Aurora MySQL Version 1 ..... 233
Compatibilité avec MySQL 5.7 ..... 234
Mises à jour d'Aurora MySQL : 15/08/2023 (version 2.07.10) (obsolète) ..... 234
Améliorations ..... 235
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 236
Compatibilité avec MySQL 5.7 ..... 236
Mises à jour d'Aurora MySQL : 04/05/2023 (version 2.07.9) (obsolète) ..... 237
Améliorations ..... 238
Fonctions non prises en charge dans Aurora MySQL version 2 ..... 239
Compatibilité avec MySQL 5.7 ..... 239
Mises à jour d'Aurora MySQL : 16/06/2022 (version 2.07.8) (obsolète) ..... 240
Améliorations ..... 241
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 241
Comparaison avec Aurora MySQL Version 1 ..... 241
Compatibilité avec MySQL 5.7 ..... 242
Mises à jour d'Aurora MySQL : 24/11/2021 (version 2.07.7) (obsolète) ..... 243

Améliorations ..... 243
Comparaison avec Aurora MySQL Version 1 ..... 244
Compatibilité avec MySQL 5.7 ..... 245
Mises à jour d'Aurora MySQL : 02/09/2021 (version 2.07.6) (obsolète) ..... 246
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 246
Comparaison avec Aurora MySQL Version 1 ..... 246
Compatibilité avec MySQL 5.7 ..... 247
Mises à jour d'Aurora MySQL : 06/07/2021 (version 2.07.5) (obsolète) ..... 248
Améliorations ..... 248
Comparaison avec Aurora MySQL Version 1 ..... 249
Compatibilité avec MySQL 5.7 ..... 249
Mises à jour d'Aurora MySQL : 04/03/2021 (version 2.07.) (obsolète) ..... 250
Améliorations ..... 251
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 252
Comparaison avec Aurora MySQL Version 1 ..... 252
Compatibilité avec MySQL 5.7 ..... 253
Mises à jour d'Aurora MySQL : 10/11/2020 (version 2.07.3) (obsolète) ..... 253
Améliorations ..... 254
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 255
Comparaison avec Aurora MySQL Version 1 ..... 257
Compatibilité avec MySQL 5.7 ..... 257
Mises à jour d'Aurora MySQL : 17/04/2020 (version 2.07.2) (obsolète) ..... 258
Améliorations ..... 259
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 260
Comparaison avec Aurora MySQL Version 1 ..... 260
Compatibilité avec MySQL 5.7 ..... 261
Mises à jour d'Aurora MySQL : 23/12/2019 (version 2.07.1) (obsolète) ..... 261
Améliorations ..... 262
Comparaison avec Aurora MySQL version 1 ..... 263
Compatibilité avec MySQL 5.7 ..... 263
Mises à jour d'Aurora MySQL : 25/11/2019 (version 2.07.0) (obsolète) ..... 264
Améliorations ..... 265
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 266
Comparaison avec Aurora MySQL Version 1 ..... 266
Compatibilité avec MySQL 5.7 ..... 267
Mises à jour d'Aurora MySQL : 22/11/2019 (version 2.06.0) (obsolète) ..... 267

Améliorations ..... 268
Comparaison avec Aurora MySQL Version 1 ..... 271
Compatibilité avec MySQL 5.7 ..... 272
Mises à jour d'Aurora MySQL : 11/11/2019 (version 2.05.0) (obsolète) ..... 272
Améliorations ..... 273
Intégration de correctifs de bogues MySQL. ..... 274
Comparaison avec Aurora MySQL Version 1 ..... 274
Compatibilité avec MySQL 5.7 ..... 275
Mises à jour d'Aurora MySQL : 14/08/2020 (version 2.04.9) (obsolète) ..... 275
Améliorations ..... 276
Intégration de correctifs de bogues MySQL. ..... 279
Comparaison avec Aurora MySQL Version 1 ..... 280
Compatibilité avec MySQL 5.7 ..... 281
Mises à jour d'Aurora MySQL : 20/11/2019 (version 2.04.8) (obsolète) ..... 281
Améliorations ..... 282
Comparaison avec Aurora MySQL Version 1 ..... 283
Compatibilité avec MySQL 5.7 ..... 283
Mises à jour d'Aurora MySQL : 14/11/2019 (version 2.04.7) (obsolète) ..... 284
Améliorations ..... 285
Comparaison avec Aurora MySQL Version 1 ..... 286
Compatibilité avec MySQL 5.7 ..... 286
Mises à jour d'Aurora MySQL : 19/09/2019 (version 2.04.6) (obsolète) ..... 287
Améliorations ..... 288
Intégration de correctifs de bogues MySQL. ..... 288
Comparaison avec Aurora MySQL Version 1 ..... 288
Compatibilité avec MySQL 5.7 ..... 289
Mises à jour d'Aurora MySQL : 08/07/2019 (version 2.045) (obsolète) ..... 290
Améliorations ..... 290
Comparaison avec Aurora MySQL Version 1 ..... 291
Compatibilité avec MySQL 5.7 ..... 292
Mises à jour d'Aurora MySQL : 29/05/2019 (version 2.04.4) (obsolète) ..... 293
Améliorations ..... 294
Comparaison avec Aurora MySQL Version 1 ..... 294
Compatibilité avec MySQL 5.7 ..... 295
Mises à jour d'Aurora MySQL : 09/05/2019 (version 2.04.3) (obsolète) ..... 295
Améliorations ..... 296

Comparaison avec Aurora MySQL Version 1 ..... 297
Compatibilité avec MySQL 5.7 ..... 297
Mises à jour d'Aurora MySQL : 02/05/2019 (version 2.04.2) (obsolète) ..... 298
Améliorations ..... 299
Intégration de correctifs de bogues MySQL. ..... 299
Comparaison avec Aurora MySQL Version 1 ..... 299
Compatibilité avec MySQL 5.7 ..... 300
Mises à jour d'Aurora MySQL : 25/03/2019 (version 2.04.1) (obsolète) ..... 301
Améliorations ..... 302
Comparaison avec Aurora MySQL Version 1 ..... 302
Compatibilité avec MySQL 5.7 ..... 302
Mises à jour d'Aurora MySQL : 25/03/2019 (version 2.04.0) (obsolète) ..... 303
Améliorations ..... 304
Intégration de correctifs de bogues MySQL. ..... 304
Comparaison avec Aurora MySQL Version 1 ..... 305
Compatibilité avec MySQL 5.7 ..... 305
Mises à jour d'Aurora MySQL : 07/02/2019 (version 2.03.4) (obsolète) ..... 306
Améliorations ..... 307
Comparaison avec Aurora MySQL Version 1 ..... 307
Compatibilité avec MySQL 5.7 ..... 308
Mises à jour d'Aurora MySQL : 18/01/2019 (version 2.03.3) (obsolète) ..... 308
Améliorations ..... 309
Intégration de correctifs de bogues MySQL. ..... 310
Comparaison avec Aurora MySQL Version 1 ..... 310
Compatibilité avec MySQL 5.7 ..... 311
Mises à jour d'Aurora MySQL : 09/01/2019 (version 2.03.2) (obsolète) ..... 312
Améliorations ..... 313
Comparaison avec Aurora MySQL Version 1 ..... 313
Compatibilité avec MySQL 5.7 ..... 313
Mises à jour d'Aurora MySQL : 24/10/2018 (version 2.03.1) (obsolète) ..... 314
Améliorations ..... 315
Comparaison avec Aurora MySQL Version 1 ..... 315
Compatibilité avec MySQL 5.7 ..... 316
Mises à jour d'Aurora MySQL : 11/10/2018 (version 2.03) (obsolète) ..... 316
Améliorations ..... 317
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 318

Comparaison avec Aurora MySQL Version 1 ..... 318
Compatibilité avec MySQL 5.7 ..... 319
Mises à jour d'Aurora MySQL : 08/10/2018 (version 2.02.5) (obsolète) ..... 319
Améliorations ..... 320
Comparaison avec Aurora MySQL Version 1 ..... 320
Compatibilité avec MySQL 5.7 ..... 321
Mises à jour d'Aurora MySQL : 21/09/2018 (version 2.02.4) (obsolète) ..... 322
Améliorations ..... 322
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 322
Comparaison avec Aurora MySQL Version 1 ..... 323
Compatibilité avec MySQL 5.7 ..... 324
Mises à jour d'Aurora MySQL : 23/08/2018 (version 2.02.3) (obsolète) ..... 325
Comparaison avec Aurora MySQL Version 1 ..... 326
Compatibilité avec MySQL 5.7 ..... 326
Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x ..... 327
Améliorations ..... 327
Mises à jour d'Aurora MySQL : 04/06/2018 (version 2.02.2) (obsolète) ..... 328
Améliorations ..... 329
Comparaison avec Aurora MySQL 5.6 ..... 329
Compatibilité avec MySQL 5.7 ..... 330
Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x ..... 327
Améliorations ..... 329
Mises à jour d'Aurora MySQL : 03/05/2018 (version 2.02) (obsolète) ..... 331
Comparaison avec Aurora MySQL 5.6 ..... 332
Compatibilité avec MySQL 5.7 ..... 332
Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x ..... 327
Améliorations ..... 333
Intégration de correctifs de bogues MySQL. ..... 334
Mises à jour d'Aurora MySQL : 13/03/2018 (version 2.01.1) (obsolète) ..... 334
Comparaison avec Aurora MySQL 5.6 ..... 335
Compatibilité avec MySQL 5.7 ..... 336
Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x ..... 336
Améliorations ..... 337
Mises à jour d'Aurora MySQL : 06/02/2018 (version 2.01) (obsolète) ..... 337
Comparaison avec Aurora MySQL 5.6 ..... 337
Compatibilité avec MySQL 5.7 ..... 338

Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x ..... 327
Amazon Aurora MySQL version 1 (obsolète) ..... 340
Mises à jour d'Aurora MySQL : 30/09/2021 (version 1.23.4) (obsolète) ..... 343
Améliorations ..... 343
Mises à jour d'Aurora MySQL : 28/06/2021 (version 1.23.3) (obsolète) ..... 344
Améliorations ..... 344
Mises à jour d'Aurora MySQL : 18/03/2021 (version 1.23.2) (obsolète) ..... 344
Améliorations ..... 345
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 347
Mises à jour d'Aurora MySQL : 24/11/2020 (version 1.23.1) (obsolète) ..... 347
Améliorations ..... 347
Mises à jour d'Aurora MySQL : 02/09/2020 (version 1.23.0) (obsolète) ..... 348
Améliorations ..... 349
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 353
Mises à jour d'Aurora MySQL : 03/06/2021 (version 1.22.5) (obsolète) ..... 354
Améliorations ..... 355
Mises à jour d'Aurora MySQL : 04/02/2021 (version 1.22.4) (obsolète) ..... 355
Améliorations ..... 356
Mises à jour d'Aurora MySQL : 19/11/2020 (version 1.22.3) (obsolète) ..... 357
Améliorations ..... 358
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 359
Mises à jour d'Aurora MySQL : 05/03/2020 (version 1.22.2) (obsolète) ..... 360
Améliorations ..... 360
Mises à jour d'Aurora MySQL : 23/12/2019 (version 1.22.1) (obsolète) ..... 361
Améliorations ..... 362
Mises à jour d'Aurora MySQL : 25/11/2019 (version 1.22.0) (obsolète) ..... 362
Améliorations ..... 363
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 367
Mises à jour d'Aurora MySQL : 25/11/2019 (version 1.21.0) (obsolète) ..... 367
Améliorations ..... 368
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 369
Mises à jour d'Aurora MySQL : 05/03/2020 (version 1.20.1) (obsolète) ..... 370
Améliorations ..... 370
Mises à jour d'Aurora MySQL : 11/11/2019 (version 1.20.0) (obsolète) ..... 371
Améliorations ..... 371
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 373

Mises à jour d'Aurora MySQL : 05/03/2020 (version 1.19.6) (obsolète) ..... 373
Améliorations ..... 374
Mises à jour d'Aurora MySQL : 19/09/2019 (version 1.19.5) (obsolète) ..... 374
Améliorations ..... 375
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 376
Mises à jour d'Aurora MySQL : 05/06/2019 (version 1.19.2) (obsolète) ..... 376
Améliorations ..... 377
Mises à jour d'Aurora MySQL : 09/05/2019 (version 1.19.1) (obsolète) ..... 377
Améliorations ..... 378
Mises à jour d'Aurora MySQL : 07/02/2019 (version 1.19.0) (obsolète) ..... 379
Fonctionnalités ..... 379
Améliorations ..... 380
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 381
Mises à jour d'Aurora MySQL : 20/09/2018 (version 1.18.0) (obsolète) ..... 381
Fonctionnalités ..... 382
Mises à jour d'Aurora MySQL : 05/03/2020 (version 1.17.9) (obsolète) ..... 383
Améliorations ..... 384
Mises à jour d'Aurora MySQL : 17/01/2019 (version 1.17.8) (obsolète) ..... 384
Améliorations ..... 385
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 385
Mises à jour d'Aurora MySQL : 08/10/2018 (version 1.17.7) (obsolète) ..... 385
Améliorations ..... 386
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 386
Mises à jour d'Aurora MySQL : 06/09/2018 (version 1.17.6) (obsolète) ..... 386
Améliorations ..... 387
Intégration de correctifs de bogues de l'édition MySQL Community Edition ..... 388
Mises à jour d'Aurora MySQL : 14/08/2018.(version 1.17.5) (obsolète) ..... 388
Améliorations ..... 389
Mises à jour d'Aurora MySQL : 07/08/2018 (version 1.17.4) (obsolète) ..... 389
Améliorations ..... 389
Mises à jour d'Aurora MySQL : 05/06/2018 (version 1.17.3) (obsolète) ..... 390
Améliorations ..... 391
Mises à jour d'Aurora MySQL : 27/04/2018 (version 1.17.2) (obsolète) ..... 391
Améliorations ..... 392
Mises à jour d'Aurora MySQL : 23/03/2018 (version 1.17.1) (obsolète) ..... 392
Améliorations ..... 393

Mises à jour d'Aurora MySQL : 13/03/2018 (version 1.17) (obsolète) ..... 393
Application de correctifs sans temps d'arrêt ..... 394
Nouvelles fonctionnalités ..... 394
Améliorations ..... 394
Intégration de correctifs de bogues MySQL. ..... 395
Mises à jour d'Aurora MySQL : 11/12/2017 (version 1.16) (obsolète) ..... 395
Application de correctifs sans temps d'arrêt ..... 396
Nouvelles fonctionnalités ..... 396
Améliorations ..... 396
Intégration de correctifs de bogues MySQL. ..... 397
Mises à jour d'Aurora MySQL : 20/11/2017 (version 1.15.1) (obsolète) ..... 397
Application de correctifs sans temps d'arrêt ..... 398
Améliorations ..... 398
Intégration de correctifs de bogues MySQL. ..... 398
Mises à jour d'Aurora MySQL : 24/10/2017 (version 1.15) (obsolète) ..... 398
Application de correctifs sans temps d'arrêt ..... 399
Nouvelles fonctionnalités ..... 399
Améliorations ..... 400
Intégration de correctifs de bogues MySQL. ..... 398
Mises à jour d'Aurora MySQL : 13/03/2018 (version 1.14.4) (obsolète) ..... 402
Application de correctifs sans temps d'arrêt ..... 402
Nouvelles fonctionnalités ..... 402
Améliorations ..... 402
Intégration de correctifs de bogues MySQL. ..... 403
Mises à jour d'Aurora MySQL : 22/09/2017 (version 1.14.1) (obsolète) ..... 403
Améliorations ..... 403
Mises à jour d'Aurora MySQL : 07/08/2017 (version 1.14) (obsolète) ..... 404
Application de correctifs sans temps d'arrêt ..... 404
Améliorations ..... 404
Intégration de correctifs de bogues MySQL. ..... 406
Mises à jour d'Aurora MySQL : 15/05/2017 (version 1.13) (obsolète) ..... 406
Application de correctifs sans temps d'arrêt ..... 406
Nouvelles fonctions ..... 407
Améliorations ..... 407
Intégration de correctifs de bogues MySQL. ..... 408
Mises à jour d'Aurora MySQL : 05/04/2017 (version 1.12) (obsolète) ..... 409

Nouvelles fonctionnalités ..... 409
Améliorations ..... 409
Intégration de correctifs de bogues MySQL. ..... 410
Mises à jour d'Aurora MySQL du 23/02/2017 (version 1.11) (obsolète) ..... 411
Nouvelles fonctions ..... 411
Améliorations ..... 412
Intégration de correctifs de bogues MySQL. ..... 414
Mises à jour d'Aurora MySQL : 12/01/2017 (version 1.10.1) (obsolète) ..... 414
Nouvelles fonctions ..... 415
Améliorations ..... 415
Mises à jour d'Aurora MySQL : 14/12/2016 (version 1.10) (obsolète) ..... 415
Nouvelles fonctions ..... 415
Améliorations ..... 417
Intégration de correctifs de bogues MySQL. ..... 418
Mises à jour d'Aurora MySQL : 10/11/2016 (versions 1.9.0, 1.9.1) (obsolète) ..... 418
Nouvelles fonctions ..... 418
Améliorations ..... 419
Mises à jour d'Aurora MySQL : 26/10/2016 (version 1.8.1) (obsolète) ..... 419
Améliorations ..... 419
Intégration de correctifs de bogues MySQL. ..... 420
Mises à jour d'Aurora MySQL : 18/10/2016 (version 1.8) (obsolète) ..... 420
Nouvelles fonctionnalités ..... 420
Améliorations ..... 421
Intégration de correctifs de bogues MySQL. ..... 422
Mises à jour d'Aurora MySQL : 20/09/2016 (version 1.7.1) (obsolète) ..... 422
Améliorations ..... 422
Mises à jour d'Aurora MySQL : 30/08/2016 (version 1.7.0) (obsolète) ..... 423
Nouvelles fonctions ..... 423
Améliorations ..... 423
Intégration de correctifs de bogues MySQL. ..... 424
Mises à jour d'Aurora MySQL : 01/06/2016 (version 1.6.5) (obsolète) ..... 424
Nouvelles fonctions ..... 424
Améliorations ..... 425
Intégration de correctifs de bogues MySQL. ..... 425
Mises à jour d'Aurora MySQL : 06/04/2016 (version 1.6) (obsolète) ..... 425
Nouvelles fonctions ..... 425

Améliorations ..... 427
Intégration de correctifs de bogues MySQL. ..... 428
Mises à jour d'Aurora MySQL : 11/01/2016 (version 1.5) (obsolète) ..... 428
Améliorations ..... 428
Intégration de correctifs de bogues MySQL. ..... 429
Mises à jour d'Aurora MySQL : 03/12/2015 (version 1.4) (obsolète) ..... 429
Nouvelles fonctions ..... 429
Améliorations ..... 430
Intégration de correctifs de bogues MySQL. ..... 430
Mises à jour d'Aurora MySQL : 16/10/2015 (versions 1.2, 1.3) (obsolète) ..... 431
Correctifs ..... 431
Améliorations ..... 432
Intégration de correctifs de bogues MySQL. ..... 432
Mises à jour d'Aurora MySQL du 24/08/2015 (version 1.1) (obsolète) ..... 435
Bogues MySQL corrigés par les mises à jour d'Aurora MySQL ..... 437
Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x ..... 437
Bogues MySQL corrigés par les mises à jour d'Aurora MySQL 2.x ..... 457
Bogues MySQL corrigés par les mises à jour d'Aurora MySQL 1.x ..... 483
Vulnérabilités de sécurité corrigées dans Aurora MySQL ..... 505
Historique de la documentation ..... 514
..... dxxxiii

# Notes de mise à jour d'Amazon Aurora Édition compatible avec MySQL

Amazon Aurora Édition compatible avec MySQL est mise à jour régulièrement. Ces mises à jour sont appliquées aux clusters de base de données Aurora MySQL lors des fenêtres de maintenance du système. Le moment où les mises à jour sont appliquées dépend du paramètre Région AWS et de la fenêtre de maintenance du cluster de base de données, ainsi que du type de mise à jour.

Les versions d'Amazon Aurora MySQL sont mises à la disposition de tous Régions AWS pendant plusieurs jours. Certaines régions peuvent afficher temporairement une version du moteur qui n'est pas encore disponible dans une autre région. Les notes de publication sont publiées une fois que les nouvelles versions sont disponibles dans toutes les régions.

Les mises à jour sont appliquées simultanément à toutes les instances d'un cluster de bases de données. Une mise à jour requiert un redémarrage de la base de données sur toutes les instances d'un cluster de bases de données. Vous rencontrerez 20 à 30 secondes d'indisponibilité, après quoi vous pourrez reprendre l'utilisation de votre ou de vos clusters de base de données. Vous pouvez consulter ou modifier vos paramètres de créneau de maintenance dans AWS Management Console.

Rubriques


- Calendriers de publication pour Amazon Aurora MySQL
- Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 3
- Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 2.
- Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 1 (obsolète)
- Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL
- Vulnérabilités de sécurité corrigées dans Aurora MySQL

# Calendriers de publication pour Amazon Aurora MySQL

Les calendriers de publication présentés sur cette page peuvent vous aider à planifier les mises à niveau des versions majeures et mineures. Pour plus d'informations sur les mises à niveau, le versionnement et le cycle de vie d'Amazon Aurora, consultez les versions d'Amazon Aurora.

Rubriques


- Calendrier des versions majeures d'Aurora MySQL
- Calendrier des versions mineures d'Aurora MySQL


## Calendrier des versions majeures d'Aurora MySQL

Les versions majeures d'Aurora MySQL sont disponibles dans le cadre du support standard au moins jusqu'à la fin du cycle de vie de la communauté pour la version communautaire correspondante. Vous pouvez continuer à exécuter une version majeure après la date de fin du support standard Aurora moyennant des frais. Pour plus d'informations, consultez les sections Utilisation d'Amazon RDS Extended Support et tarification d'Amazon Aurora.

Aurora MySQL prend actuellement en charge les versions principales suivantes.

> **Note**
>
> Vous pouvez également consulter les informations relatives aux dates de support des principales versions du moteur à l'aide de l'API AWS CLI ou RDS. Pour plus d'informations, consultez Affichage des dates de support pour les versions de moteur dans Support étendu Amazon RDS.

| Version majeure communautaire | Version majeure d'Aurora | Date de fin de vie de la communauté | Date de fin du support standard Aurora | Date de début de la première année de tarification du support étendu RDS | Date de début de la troisième année de tarification du support étendu RDS | Date de fin du support étendu RDS | Versions mineures éligibles au support étendu RDS |
|---|---|---|---|---|---|---|---|
| MySQL 8.0 | Aurora MySQL version 3 | Avril 2026 | 30 avril 2028 | 1er mai 2028 | Non applicable | 31 juillet 20 | À déterminer |
| MySQL 5.7 | Aurora MySQL version 2 | Octobre 20 | 31 octobre | 1er décemb | Non applicable | 28 février 2 | Aurora MySQL 2.11 et 2.12 |
| MySQL 5.6 (obsolète) | Aurora MySQL version 1 (obsolète) | 5 février 20 | 28 février 2 | Non applicable | Ne s'applique pas | Ne s'applique pas | Non applicable |

> **Note**
>
> Le support étendu Amazon RDS pour Aurora MySQL version 2 commence le 1er novembre 2024, mais ne sera pas facturé avant le 1er décembre 2024. Entre le 1er et le 30 novembre 2024, tous les clusters de bases de données Aurora MySQL version 2 sont couverts par le support étendu Amazon RDS.

# Calendrier des versions mineures d'Aurora MySQL

Aurora MySQL prend actuellement en charge les versions mineures suivantes.

En général, les versions mineures d'Aurora sont publiées tous les trimestres. Le calendrier de publication peut varier selon la nécessité d'intégrer des fonctions ou correctifs supplémentaires.

Les versions mineures peuvent atteindre la fin du support standard avant les versions majeures correspondantes. Par exemple, la version 3.07 atteindra sa date de fin de support standard en août 2025, tandis que la version majeure 3 atteindra sa fin de support standard le 30 avril 2028. Amazon RDS prendra en charge les versions mineures 3.* supplémentaires publiées entre ces dates.

Les frais de support étendu Amazon RDS s'appliquent uniquement à certaines versions mineures une fois qu'une version majeure est éligible au support étendu. Pour plus d'informations sur les versions majeures éligibles au Support étendu, consultezCalendrier des versions majeures d'Aurora MySQL.

| Aurora MySQL Version | Date de sortie d'Aurora MySQL | Date de fin du support standard pour Aurora MySQL |
|---|---|---|
| 3.10 (Compatible avec Community MySQL 8.0.42) (LTS) | 31 juillet 2025 | 30 avril 2028 |
| 3.09 (Compatible avec Community MySQL 8.0.40) | 14 mai 2025 | 14 mai 2026 |
| 3.08 (Compatible avec Community MySQL 8.0.39) | 18 novembre 2024 | 31 mars 2026 |
| 3.04 (Compatible avec Community MySQL 8.0.28) (LTS) | 31 juillet 2023 | 31 octobre 2026 |
| 2.12 [1] (Compatible avec Community MySQL 5.7.40 ou 5.7.44 [2]) | 25 juillet 2023 | 31 octobre 2024 |
| 2.11 [1] (Compatible avec Community MySQL 5.7.12) | 25 octobre 2022 | 31 octobre 2024 |

LTS — Versions de support à long terme (LTS) d'Aurora MySQL. Pour plus d'informations, consultez Support à long terme (LTS) et versions bêta d'Amazon Aurora MySQL.

[1] Cette version mineure continuera d'être disponible lorsque la version majeure sera disponible sur Amazon RDS Extended Support.

[2] Les versions 2.12 à 2.12.1 d'Aurora MySQL sont compatibles avec MySQL version 5.7.40, et les versions 2.12.2 et supérieures sont compatibles avec MySQL version 5.7.44.

# Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 3

Les mises à jour du moteur de base de données Amazon Aurora version 3 sont les suivantes.

Rubriques


- Mises à jour du moteur de base de données Aurora MySQL 2025-11-13 (version 3.11.0, compatible avec MySQL 8.0.43)
- Mises à jour du moteur de base de données Aurora MySQL 2025-11-20 (version 3.10.2, compatible avec MySQL 8.0.42)
- Mises à jour du moteur de base de données Aurora MySQL 2025-09-30 (version 3.10.1, compatible avec MySQL 8.0.42)
- Mises à jour du moteur de base de données Aurora MySQL 2025-07-31 (version 3.10.0, compatible avec MySQL 8.0.42)
- Mises à jour du moteur de base de données Aurora MySQL 14/05/2025-05 (version 3.09.0, compatible avec MySQL 8.0.40)
- Mises à jour du moteur de base de données Aurora MySQL 2025-04-07 (version 3.08.2, compatible avec MySQL 8.0.39)
- Mises à jour du moteur de base de données Aurora MySQL 2025-02-17 (version 3.08.1, compatible avec MySQL 8.0.39)
- Mises à jour du moteur de base de données Aurora MySQL 2024-11-18 (version 3.08.0, compatible avec MySQL 8.0.39)
- Mises à jour du moteur de base de données Aurora MySQL 2024-07-23 (version 3.07.1, compatible avec MySQL 8.0.36)
- Mises à jour du moteur de base de données Aurora MySQL 2024-06-04 (version 3.07.0, compatible avec MySQL 8.0.36)
- Mises à jour du moteur de base de données Aurora MySQL 2024-06-26 (version 3.06.1, compatible avec MySQL 8.0.34)
- Mises à jour du moteur de base de données Aurora MySQL 2024-03-07 (version 3.06.0, compatible avec MySQL 8.0.34)
- Mises à jour du moteur de base de données Aurora MySQL 2024-01-31 (version 3.05.2, compatible avec MySQL 8.0.32)

- Mises à jour du moteur de base de données Aurora MySQL 21/11/2023 (version 3.05.1, compatible avec MySQL 8.0.32)
- Mises à jour du moteur de base de données Aurora MySQL du 30/10/2023 (version 3.05.0.1, compatible avec MySQL 8.0.32), Bêta
- Mises à jour du moteur de base de données Aurora MySQL du 25/10/2023 (version 3.05.0, compatible avec MySQL 8.0.32)
- Mises à jour du moteur de base de données Aurora MySQL 2025-05-05 (version 3.04.4, compatible avec MySQL 8.0.28)
- Mises à jour du moteur de base de données Aurora MySQL 2024-06-26 (version 3.04.3, compatible avec MySQL 8.0.28)
- Mises à jour du moteur de base de données Aurora MySQL 2024-03-15 (version 3.04.2, compatible avec MySQL 8.0.28)
- Mises à jour du moteur de base de données Aurora MySQL 13/11/2023 (version 3.04.1, compatible avec MySQL 8.0.28)
- Mises à jour du moteur de base de données Aurora MySQL 31/07/2023 (version 3.04.0, compatible avec MySQL 8.0.28)
- Mises à jour du moteur de base de données Aurora MySQL 08/2023 (version 3.03.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL 29/08/2023 (version 3.03.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL 11/05/2023 (version 3.03.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 01/01 (version 3.03.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL 17/04/2023 (version 3.02.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL le 18 novembre 2018 (version 3.02.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL 07.09.07 (version 3.02.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL 20/04/2020 (version 3.02.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL 15/04/15 (version 3.01.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL 2021-11-18 (version 3.01.0) (obsolète)

# Mises à jour du moteur de base de données Aurora MySQL 2025-11-13 (version 3.11.0, compatible avec MySQL 8.0.43)

Version : 3.11.0

Aurora MySQL 3.11.0 est généralement disponible. Les versions 3.11 d'Aurora MySQL sont compatibles avec MySQL 8.0.43. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Vous pouvez effectuer une mise à niveau sur place en utilisant des correctifs sans interruption de service (ZDP), restaurer un instantané ou lancer une mise à niveau gérée à blue/green l'aide des Blue/Green déploiements Amazon RDS depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.11.0.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez la section Mise à niveau des clusters de bases de données Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, elle Support est disponible sur les forums communautaires et via Support. Pour plus d'informations, consultez la section Maintenance d'un cluster de base de données Aurora dans le guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité résolus et CVEs

Moyen CVEs :

- CVE-2025-50077
- CVE-2025-50078
- CVE-2025-50079
- CVE-2025-50080

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

Faible CVEs :

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

Améliorations de disponibilité

- Correction d'un problème lié à la restauration logique lorsque cette `aurora_enhanced_binlog` option était activée et qui empêchait le redémarrage de la base de données en cas d'interruption de transactions.
- Correction d'un problème qui pouvait entraîner le redémarrage de l'instance du rédacteur lorsque le transfert d'écriture global ou le transfert d'écriture local était désactivé.

- Correction d'un problème qui empêchait Writer de répondre lorsque le transfert d'écriture était désactivé alors que deux répliques ou plus transféraient des demandes.
- Correction d'un problème qui pouvait provoquer le redémarrage du moteur lors de `KILL <query-id>` son exécution après `EXPLAIN FOR CONNECTION <query-id>` l'exécution d'une requête parallèle.
- Correction d'un problème qui out-of-memory empêchait (OOM) de conserver la valeur des paramètres de base de données configurée après le `aurora_oom_response` redémarrage de la base de données.
- Correction d'un problème qui empêchait les utilisateurs disposant de `SUPER` droits `CONNECTION_ADMIN` ou de privilèges d'établir une connexion supplémentaire au-delà des `max_connections` limites fixées, comme le permettait MySQL Community Edition.
- Correction d'un problème lié au dimensionnement d'Aurora Serverless V2 qui entraînait le redémarrage des instances de base de données en empêchant le remplacement de pages mémoire critiques.
- Correction d'un problème qui pouvait entraîner le redémarrage des instances Aurora MySQL Serverless v2 lorsque le `innodb_purge_threads` paramètre était configuré manuellement à une valeur différente de la valeur par défaut. Le `innodb_purge_threads` paramètre est désormais géré automatiquement pour les instances Aurora Serverless v2 et ne peut pas être modifié.
- Correction d'un problème qui pouvait entraîner l'échec du redémarrage d'une instance de lecteur lorsque le rédacteur exécutait un grand nombre d'opérations DDL.
- Correction d'un problème qui pouvait empêcher l'instance du rédacteur de répondre si les instances du lecteur redémarraient lors de l'utilisation du transfert d'écriture global ou du transfert d'écriture local.
- Correction d'un problème qui pouvait provoquer le redémarrage du moteur lors de l'application de correctifs sans interruption (ZDP) ou du redémarrage sans interruption de service (ZDR), lors de la préservation des connexions avec des transactions actives. SSL/TLS
- Correction d'un problème qui, dans de rares cas, pouvait entraîner l'envoi des modifications apportées à l'instance du rédacteur à l'instance du lecteur dans un ordre incompatible avec les threads de traitement des requêtes. Cela peut entraîner un blocage entre le thread de traitement des requêtes et le thread d'application de réplication, ce qui peut à son tour provoquer le redémarrage de l'instance du lecteur.
- Correction d'un problème qui empêchait les redémarrages inutiles du serveur de base de données dus à une évaluation incorrecte de l'état de santé de l'agent de surveillance.

### Améliorations générales

- Correction d'un problème en raison duquel les écritures dans la base de données pouvaient se bloquer lors de l'exécution d'une transaction de longue durée entraînant le redémarrage de la base de données ou l'échec d'une mise à niveau d'une version majeure.
- Correction d'un problème en raison duquel les répliques ne redémarraient pas correctement lorsque vous rejoigniez le rédacteur.
- Performances de base de données optimisées pour les instances de base de données dotées de grands volumes de stockage Aurora.
- Correction d'un problème dans l'optimiseur de plage en raison duquel les expressions constantes n'étaient pas reconnues lors du calcul des limites de plage. (Bug communautaire #112737)
- Correction d'un problème en raison duquel l'exportation d'un instantané de base de données vers Amazon S3 était plus lente lors de l'exportation de tables contenant des colonnes JSON avec des valeurs nulles.
- Correction d'un problème en raison duquel les fichiers journaux binaires temporaires n'étaient pas correctement nettoyés après l'annulation des transactions lors de l'utilisation de la journalisation binaire. Ce correctif évite la consommation de stockage liée aux fichiers temporaires conservés et, dans certains cas, peut également empêcher les anomalies dans les fichiers journaux binaires.
- Amélioration des performances d'IOPS en écriture lorsque la variable système `innodb_flush_log_at_trx_commit` est définie sur. `0`
- Correction d'un problème selon lequel l'application de correctifs sans interruption (ZDP) ou le redémarrage sans interruption (ZDR) pouvaient entraîner le redémarrage de l'instance de base de données lors de la restauration des avertissements contenant des codes d'erreur non valides.
- Désactive automatiquement `aurora_oom_response` les actions (sauf l'impression, si elle est configurée`aurora_oom_response`) lorsque la pression de mémoire ne peut être résolue après un certain temps (de l'ordre de quelques minutes).
- Correction d'un problème en raison duquel une colonne présentant des mises à jour JSON partielles échouait à être exportée en parallèle, ce qui provoquait un retour interne à une exportation Amazon RDS beaucoup plus lente.
- Correction d'un problème qui pouvait entraîner des problèmes de gestion de la mémoire lors d'opérations de requête parallèles sur la table contenant les champs blob.
- Correction d'un problème qui pouvait entraîner l'exclusion de certains GTIDs événements `Previous_gtids` binlog lorsque Enhanced Binlog était activé et `gtid_mode` défini sur ou. `ON` `ON_PERMISSIVE`

- Correction d'un problème qui provoquait une « erreur interne de transfert d'écriture » inattendue sur les instances du lecteur lorsque le transfert d'écriture était activé.
- Résolution d'un problème de course qui pouvait entraîner des lectures de pages incorrectes depuis le pool de mémoire tampon pendant les opérations de réduction d'Aurora Serverless ou lors de l'expulsion de pages du pool de mémoire tampon. Voir le bogue #116305 de la communauté.
- Problème résolu : l'exportation d'un instantané de base de données vers Amazon S3 était plus lente lors de l'exportation de tables contenant des index secondaires avec des expressions générées.
- Correction d'un problème qui provoquait des `AbortedClients` statistiques inexactes lorsque plusieurs connexions se terminaient de manière inattendue.
- Ajout d'une nouvelle variable globale `aurora_lambda_request_timeout` pour permettre aux utilisateurs de configurer les délais AWS Lambda d'expiration des demandes (par défaut : 10 secondes). Pour plus d'informations sur l'appel d'une fonction Lambda depuis un cluster de bases de données Aurora MySQL, consultez Invoquer une fonction Lambda depuis un cluster de base de données Aurora MySQL.
- Correction d'un problème en raison duquel le transfert d'écriture local cessait de fonctionner après le redémarrage de l'instance de base de données sans interruption de service.
- Correction d'un problème qui pouvait entraîner l'échec d'une opération de redémarrage d'une instance de base de données si elle `max_user_connections` était définie sur une faible valeur.
- Performances d'exportation parallèle améliorées en optimisant le processus d'amorçage pour les gros volumes (> 64 To), réduisant ainsi la durée globale des opérations d'exportation.
- Correction d'un problème qui pouvait entraîner le redémarrage de l'instance du rédacteur `ALTER TABLE` en cas d'exécution parallèle de requêtes de lecture.
- Correction d'un problème en raison duquel les exportations de clusters de bases de données pouvaient prendre beaucoup plus de temps que prévu lorsque certaines tables dépassaient 14 To.
- Correction d'un problème qui provoquait un suivi inexact des demandes de requêtes parallèles lors de l'exécution d'`EXPLAIN ANALYZE`instructions `Aurora_pq_request_in_progress` pour lesquelles le compteur n'était pas correctement mis à jour.
- Correction d'un problème qui, dans des circonstances peu communes, pouvait provoquer le redémarrage de l'instance de base de données lorsque le volume de base de données était proche de sa taille maximale autorisée.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de la restauration des connexions lors du redémarrage sans interruption et de l'application de correctifs sans interruption de service.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.43 incluse. Pour de plus amples informations, veuillez consulter Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème à cause duquel une requête du formulaire `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` entraînait un échec d'assertion dans`item_func.cc`. (Correctif de bogue communautaire #36128964)
- Une requête du formulaire `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` a conduit à une assertion dans`item_func.cc`. (Insecte #36128964)

Pour plus d'informations sur les modifications apportées à la communauté, consultez les notes de mise à jour de MySQL 8.0.43.

## Mises à jour du moteur de base de données Aurora MySQL 2025-11-20 (version 3.10.2, compatible avec MySQL 8.0.42)

Version : 3.10.2

Aurora MySQL 3.10.2 est généralement disponible. Les versions 3.10 d'Aurora MySQL sont compatibles avec MySQL 8.0.42. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Vous pouvez effectuer une mise à niveau sur place en utilisant les correctifs ZDP (Zero Downtime Patching), restaurer un instantané ou lancer une mise à niveau gérée à blue/green l'aide des Blue/Green déploiements Amazon RDS depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.10.2.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour des informations générales sur les mises à niveau d'Aurora MySQL, consultez la section Mise à niveau des clusters de bases de données Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, elle Support est disponible sur les forums communautaires et via Support. Pour plus d'informations, consultez la section Maintenance d'un cluster de base de données Aurora dans le guide de l'utilisateur Amazon Aurora.

## Améliorations

Améliorations de la disponibilité :

- Correction d'un problème qui pouvait provoquer le redémarrage du moteur lors de l'application de correctifs sans interruption (ZDP) ou du redémarrage sans interruption (ZDR), lors de la préservation des connexions avec des transactions actives. SSL/TLS
- Correction d'un problème qui pouvait provoquer le redémarrage du moteur lors de `KILL <query-id>` son exécution après l'exécution `EXPLAIN FOR CONNECTION <query-id>` d'une requête parallèle en cours d'exécution.
- Correction d'un problème qui empêchait Writer de répondre lorsque le transfert d'écriture était désactivé alors que deux répliques ou plus transféraient des demandes.
- Correction d'un problème qui pouvait entraîner le redémarrage de l'instance du rédacteur lorsque le transfert d'écriture global ou le transfert d'écriture local était désactivé.

Améliorations générales :

- Correction d'un problème qui provoquait un suivi inexact des requêtes de requête parallèle lors de l'exécution d'`EXPLAIN ANALYZE`instructions lorsque le `Aurora_pq_request_in_progress` compteur n'était pas correctement mis à jour.
- Désactive automatiquement `aurora_oom_response` les actions (sauf l'impression, si elle est configurée`aurora_oom_response`) lorsque la pression de mémoire ne peut être résolue après un certain temps (de l'ordre de quelques minutes).
- Amélioration des performances d'IOPS en écriture lorsque la variable système `innodb_flush_log_at_trx_commit` est définie sur. `0`

- Correction d'un problème en raison duquel les répliques ne redémarraient pas correctement lorsque vous rejoigniez le rédacteur.
- Correction d'un problème en raison duquel les écritures dans la base de données pouvaient se bloquer lors de l'exécution d'une transaction de longue durée entraînant le redémarrage de la base de données ou l'échec d'une mise à niveau d'une version majeure.

# Mises à jour du moteur de base de données Aurora MySQL 2025-09-30 (version 3.10.1, compatible avec MySQL 8.0.42)

Version : 3.10.1

Aurora MySQL 3.10.1 est généralement disponible. Les versions 3.10 d'Aurora MySQL sont compatibles avec MySQL 8.0.42. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Vous pouvez effectuer une mise à niveau sur place en utilisant les correctifs ZDP (Zero Downtime Patching), restaurer un instantané ou lancer une mise à niveau gérée à blue/green l'aide des Blue/Green déploiements Amazon RDS depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.10.1.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez la section Maintenance d'un cluster de base de données Aurora dans le guide de l'utilisateur Amazon Aurora.

# Améliorations

Améliorations de la disponibilité :

- Correction d'un problème lié à l'ordre de verrouillage des pages envoyé aux instances du lecteur. Cela peut provoquer le redémarrage de l'instance du lecteur en raison d'un blocage lors de l'exécution de requêtes SELECT sur des tables mises à jour par son auteur.
- Correction d'un problème qui pouvait entraîner l'échec du redémarrage d'une instance de lecteur lorsque le rédacteur exécutait un grand nombre d'opérations DDL.
- Correction d'un problème qui pouvait entraîner le redémarrage des instances Aurora MySQL Serverless v2 lorsque le paramètre innodb_purge_threads était configuré manuellement à une valeur différente de la valeur par défaut. Le paramètre innodb_purge_threads est désormais géré automatiquement pour les instances Aurora Serverless v2 et ne peut pas être modifié.
- Correction d'un problème lié au dimensionnement d'Aurora Serverless V2 qui entraînait le redémarrage des instances de base de données en empêchant le remplacement de pages mémoire critiques.
- Correction d'un problème qui empêchait Aurora de ne pas conserver la valeur configurée du paramètre de base de données aurora_oom_response après le redémarrage de la base de données pour éviter le manque de mémoire (OOM).
- Correction d'un problème de concurrence qui pouvait entraîner un problème de disponibilité après un basculement dans des clusters multi-AZ.
- Correction d'un problème de restauration logique avec aurora_enhanced_binlog activé qui pouvait empêcher le redémarrage de la base de données en cas d'interruption de transactions.
- Résolution d'un problème en raison duquel les instances de base de données pouvaient ne plus répondre pendant les charges de travail des requêtes parallèles (PQ).
- Correction d'un problème qui empêchait les utilisateurs disposant des privilèges CONNECTION_ADMIN ou SUPER d'établir une connexion supplémentaire au-delà de la limite max_connections, comme le permettait MySQL Community Edition.

Améliorations générales :

- Correction d'un problème qui provoquait des redémarrages continus du serveur de base de données lorsque le volume était autorisé à atteindre sa taille maximale en raison de l'autorisation erronée de certains types de requêtes une fois que le volume était supérieur à un certain seuil.

- Correction d'un problème qui empêchait l'instance du rédacteur de répondre si les instances du lecteur redémarraient lors de l'utilisation du transfert d'écriture global ou du transfert d'écriture local.
- Correction d'un problème qui pouvait entraîner le redémarrage de l'instance du rédacteur lors de l'exécution d'ALTER TABLE en parallèle avec des requêtes de lecture.
- Correction d'un problème d'amélioration de la disponibilité lié au démarrage de gros volumes (> 64 To) lors de l'opération d'exportation parallèle.
- Correction d'un problème qui pouvait entraîner l'échec d'une opération de redémarrage d'une instance de base de données si max_user_connections était défini sur une faible valeur.
- Correction d'un problème qui provoquait des AbortedClients statistiques inexactes lorsque plusieurs connexions se terminaient de manière inattendue.
- Résolution d'un problème de course qui pouvait entraîner des lectures de pages incorrectes depuis le pool de mémoire tampon pendant les opérations de réduction d'Aurora Serverless ou lors de l'expulsion de pages du pool de mémoire tampon. Voir le bogue #116305 de la communauté.
- Correction d'un problème qui provoquait une « erreur interne de transfert d'écriture » inattendue sur les instances du lecteur lorsque le transfert d'écriture était activé.
- Correction d'un problème qui pouvait entraîner l'exclusion de certains événements binlog Previous_GTIDS lorsque Enhanced Binlog était activé et que gtid_mode GTIDs était défini sur ON ou ON_PERMISSIVE.
- Correction d'un problème qui pouvait entraîner un problème de gestion de la mémoire lors d'opérations de requête parallèles sur la table contenant les champs blob.
- Correction d'un problème selon lequel une colonne avec des mises à jour JSON partielles échouait à l'exportation parallèle, ce qui provoquait un retour interne à une exportation RDS beaucoup plus lente.
- Correction d'un problème qui pouvait provoquer des redémarrages inattendus des instances en cas de charge de travail élevée pour le transfert d'écriture simultané.
- Correction d'un problème selon lequel les correctifs ZDP (Zero Downtime Patching) et ZDR (Zero Downtime Restart) pouvaient entraîner le redémarrage de l'instance de base de données lors de la restauration des avertissements contenant des codes d'erreur non valides.
- Correction d'un problème en raison duquel les fichiers journaux binaires temporaires n'étaient pas correctement nettoyés après l'annulation des transactions lors de l'utilisation de la journalisation binaire. Ce correctif empêche la consommation inutile de stockage liée aux fichiers temporaires conservés et, dans certains cas, pourrait également empêcher des anomalies dans les fichiers journaux binaires.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires, y compris la version 8.0.42. Pour de plus amples informations, veuillez consulter Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Une requête de la forme SELECT 1 FROM t WHERE CAST (a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t) a conduit à une assertion dans item_func.cc. (Insecte #36128964)

## Mises à jour du moteur de base de données Aurora MySQL 2025-07-31 (version 3.10.0, compatible avec MySQL 8.0.42)

Version : 3.10.0

Aurora MySQL 3.10.0 est généralement disponible. Les versions 3.10 d'Aurora MySQL sont compatibles avec MySQL 8.0.42. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
> Nous vous recommandons de ne pas définir le `AutoMinorVersionUpgrade` paramètre sur `true` (ou de ne pas activer la mise à niveau automatique des versions mineures dans AWS Management Console) pour les versions LTS. Cela pourrait entraîner la mise à niveau de votre cluster de base de données vers la prochaine version cible pour la campagne de mise à niveau automatique des versions mineures, qui n'est peut-être pas une version LTS.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Vous pouvez effectuer une mise à niveau sur place en utilisant les correctifs ZDP (Zero Downtime Patching), restaurer un instantané ou lancer une mise à niveau gérée à blue/green l'aide des Blue/Green déploiements Amazon RDS depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.10.0.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez la section Maintenance d'un cluster de base de données Aurora dans le guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctionnalités

- La version 3.10 d'Aurora MySQL étend la prise en charge du cache de journaux de relais en mémoire pour les répliques de journaux binaires. Cette fonctionnalité, introduite pour la première fois dans la version 3.05, peut améliorer le débit de réplication des journaux binaires jusqu'à 40 %. Le cache de journaux de relais en mémoire est activé par défaut pour la réplication de journaux binaires à thread unique, la réplication multithread avec le positionnement automatique GTID activé, et à partir de la version 3.10, il est également activé pour la réplication multithread avec (même sans). `replica_preserve_commit_order = ON` GTIDs Vous pouvez contrôler cette fonctionnalité à l'aide d'un nouveau paramètre dans la version 3.10, `aurora_in_memory_relaylog`. Pour plus d'informations, consultez Optimisations des journaux binaires dans Aurora MySQL.
- Amazon Aurora a doublé sa capacité de stockage maximale de 128 TiB à 256 TiB, permettant ainsi des charges de travail plus importantes dans un seul cluster de base de données. Pour utiliser la limite de stockage accrue avec Aurora MySQL, mettez à niveau votre cluster vers la version 3.10 (compatible avec MySQL 8.0.42) ou une version ultérieure. Après la mise à niveau, le stockage Aurora évolue automatiquement jusqu'à 256 TiB en fonction de la quantité de données de votre volume de cluster.

# Améliorations

## Correctifs de sécurité

### Moyen CVEs :

- CVE-2025-21501
- CVE-2025-21500
- CVE-2025-21543
- CVE-2025-21540
- CVE-2025-21491
- CVE-2025-21490
- CVE-2025-21559
- CVE-2025-21555
- CVE-2025-21497
- CVE-2025-21519
- CVE-2025-21529
- CVE-2025-21505
- CVE-2025-21531
- CVE-2025-21523
- CVE-2025-21503
- CVE-2025-21522
- CVE-2025-21518
- CVE-2025-21577
- CVE-2025-30682
- CVE-2025-30687
- CVE-2025-30688
- CVE-2025-21574
- CVE-2025-21575
- CVE-2025-30693

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

Faible CVEs :

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

Améliorations de la disponibilité :

- Correction d'un problème qui provoquait des redémarrages inutiles du serveur de base de données en raison d'une évaluation incorrecte de la progression de la restauration.

Améliorations générales :

- Latence de validation réduite lorsque l' I/O optimisation est activée.

- Correction d'un problème qui empêchait les instances de lecteur de générer des journaux d'erreurs lorsque le transfert d'écriture était activé et `aurora_replica_read_consistency` que les paramètres étaient modifiés.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de lecteur en raison de l'interaction entre le thread de réplication et une requête accédant à des tables absentes du cache tampon.
- Correction d'un problème en raison duquel le transfert d'écriture local cessait de fonctionner après le redémarrage de l'instance de base de données sans interruption de service.
- Correction d'un problème de transfert d'écriture en raison duquel les requêtes transférées pouvaient se comporter de manière incorrecte pour les instructions contenant des options définies.
- Correction d'un problème de stabilité en raison duquel l'insertion de métadonnées pour un tablespace d'annulation déclenchait un redémarrage inattendu de la base de données.
- Correction d'un problème en raison duquel les exportations de clusters de bases de données pouvaient prendre beaucoup plus de temps que prévu lorsque certaines tables dépassaient 14 To.
- Correction d'un problème qui pouvait entraîner un signalement incorrect de la variable d'`Innodb_buffer_pool_pages_misc`état.
- Ajout de la prise en charge de la préservation `LAST_INSERT_ID` lors de l'application de correctifs sans interruption (ZDP) ou du redémarrage sans interruption de service (ZDR).
- Aurora MySQL utilise des valeurs 8 bits pour l'index virtuel afin IDs d'éviter les problèmes de format d'annulation de MySQL, car le dépassement de cette limite pourrait entraîner l'indisponibilité du cluster. À l'approche de cette limite, le système écrit désormais des messages d'avertissement dans le journal des erreurs MySQL. Si la limite est atteinte, les tentatives d'ajout d'un nouvel index renvoient une erreur. Pour plus d'informations sur les meilleures pratiques en matière d'index virtuel, consultez la section Erreurs de dépassement d'identifiant d'index virtuel dans la documentation d'Aurora MySQL.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires, y compris la version 8.0.42. Pour de plus amples informations, veuillez consulter Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Une sortie du serveur peut résulter de tentatives simultanées par plusieurs threads d'enregistrer et de désenregistrer des objets du schéma de performance des métadonnées, ou d'acquérir et de libérer des verrous de métadonnées. (Bogue n° 26502135)

# Mises à jour du moteur de base de données Aurora MySQL 14/05/2025-05 (version 3.09.0, compatible avec MySQL 8.0.40)

Version : 3.09.0

Aurora MySQL 3.09.0 est généralement disponible. Les versions 3.09 d'Aurora MySQL sont compatibles avec MySQL 8.0.40. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Vous pouvez effectuer une mise à niveau sur place qui exploite un cluster Aurora MySQL version zero-downtime-patch3.09.0, en restaurer un instantané ou lancer une blue/green mise à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez la section Maintenance d'un cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

## Points forts de la publication

- Bases de données globales Aurora MySQL améliorées pour permettre aux instances de lecteurs secondaires de terminer le démarrage et de traiter les demandes de lecture lors d'événements imprévus (pannes matérielles, perturbations du réseau). Auparavant, les instances de lecteur

secondaire ne pouvaient pas redémarrer lors de tels événements. Pour plus d'informations, consultez la section Résilience interrégionale pour les clusters secondaires de Global Database dans le guide de l'utilisateur Amazon Aurora.

- Réduction du temps d'arrêt des rédacteurs lors des changements interrégionaux de la base de données globale Aurora MySQL à moins d'une minute, minimisant ainsi les temps d'arrêt lors des changements régionaux planifiés.

## Améliorations

Correctifs de sécurité

Critique CVEs :

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

Moyen CVEs :

- CVE-2024-7264
- CVE-2024-21193
- CVE-2024-21194
- CVE-2024-21196
- CVE-2024-21197
- CVE-2024-21198
- CVE-2024-21199
- CVE-2024-21201
- CVE-2024-21203
- CVE-2024-21207
- CVE-2024-21212
- CVE-2024-21213
- CVE-2024-21218
- CVE-2024-21219
- CVE-2024-21230

- CVE-2024-21236
- CVE-2024-21238
- CVE-2024-21239
- CVE-2024-21241
- CVE-2025-21494
- CVE-2025-21504
- CVE-2025-21525
- CVE-2025-21534
- CVE-2025-21536

Améliorations de la disponibilité :

- Correction d'un problème en raison duquel les requêtes à instructions multiples transférées d'une instance de lecteur à une instance d'écriture pouvaient `innodb_flush_log_at_trx_commit` se bloquer lorsque cette valeur était définie `0` sur le rédacteur et non sur zéro sur le lecteur, ce qui empêchait d'éventuels échecs de transfert d'écriture.
- Correction d'un problème de blocage lié à l'activation d'Enhanced Binlog, qui pouvait entraîner le redémarrage de la base de données en cas d'exécution `SHOW BINARY LOGS` simultanée de transactions sur les moteurs BLACKHOLE ou d'exécution d'`XA PREPARE`instructions, évitant ainsi d'éventuels blocages d'écriture et problèmes de disponibilité des instances.
- Correction de problèmes de transfert d'écriture susceptibles de provoquer le redémarrage de l'instance d'écriture Aurora en empêchant l'acceptation de nouvelles demandes avant que les demandes précédentes ne soient complètement terminées, améliorant ainsi la stabilité des opérations de transfert d'écriture.
- Correction d'un problème lié à la réplication : une interruption du réseau risquait de ne pas rétablir correctement la connexion avec le rédacteur, ce qui bloquait la réplication et risquait de redémarrer l'instance.
- La réponse Aurora MySQL Out of Memory (OOM) implémente désormais le redimensionnement progressif du pool de tampons qui réduit progressivement l'utilisation de la mémoire en fonction de l'état de la mémoire du système (LOW/RESERVED) lorsqu'il est activé via le paramètre de base de `aurora_oom_response` données, offrant ainsi une meilleure gestion de la mémoire dans les situations de surcharge mémoire.
- Amélioration du temps de restauration du fichier Binlog lors du redémarrage de la base de données en optimisant le processus de restauration pour qu'il prenne un temps constant, quelle que soit la

taille du fichier Binlog. Auparavant, dans certains cas, le temps de restauration était proportionnel à la taille du dernier fichier Binlog.

- Correction d'un problème qui pouvait provoquer des redémarrages inattendus du serveur MySQL lors de l'exécution simultanée d'opérations de troncature de table InnoDB lors d'une requête. `performance_schema.data_lock_waits`
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de la validation d'événements de journal binaire volumineux dans des conditions de stockage insuffisantes.
- Correction d'un problème en raison duquel les opérations de redimensionnement du pool de mémoire tampon déclenchées en cas de manque de mémoire (OOM) pouvaient ne pas répondre dans les scénarios de charge de travail élevée, ce qui pouvait entraîner un redémarrage de la base de données.
- Correction d'un problème qui pouvait provoquer une boucle de redémarrage de la base de données lors de la création d'un déclencheur. Le problème peut également se produire lors de l'ajout d'un nouveau fichier Binlog ou Relaylog ou lors de la rotation de ces fichiers.
- Correction d'un problème qui pouvait provoquer le redémarrage de l'instance du lecteur Aurora lors de l'utilisation du transfert d'écriture avec des requêtes de validation implicites ou à instructions multiples.

Améliorations générales :

- Correction d'un problème en raison duquel les `ALTER TABLE ... REBUILD / OPTIMIZE TABLE` opérations pouvaient consommer trop de mémoire en allouant des `innodb_ddl_buffer_size` octets par thread DDL au lieu de diviser la taille de la mémoire tampon entre les threads, empêchant ainsi une éventuelle surutilisation de la mémoire pendant les opérations DDL.
- Modification de la valeur par défaut pour`aurora_oom_response`, sur toutes les classes d'instance de base de données disposant de plus de 4 GiB de mémoire, de print à print, decline, kill_connect. Pour plus d'informations, consultez la section out-of-memoryProblèmes liés à Amazon Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.
- Les privilèges suivants ont été ajoutés au `rds_superuser_role` : `FLUSH_OPTIMIZER_COSTSFLUSH_STATUS,FLUSH_TABLES,FLUSH_USER_RESOURCES`. Pour plus d'informations à ce sujet`rds_superuser_role`, consultez la documentation Amazon Master User Accounts with Amazon Aurora. Pour plus d'informations sur ces privilèges dynamiques, consultez la documentation MySQL.

- À partir de cette version d'Aurora MySQL, l'optimisation de l'insertion rapide n'est plus activée. Pour plus d'informations, consultez Améliorations des performances Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Correction d'un problème lié à un dépassement incorrect du `max_user_connections` seuil, qui provoquait des erreurs de connexion pour certains utilisateurs. Cela se produit dans certains cas extrêmes, par exemple lorsque des connexions sont créées et supprimées presque immédiatement.
- Correction d'un problème de journalisation des audits qui entraînait une utilisation élevée du processeur et empêchait l'instance de serveur de base de données de répondre.
- Correction d'un problème de gestion de la mémoire lors de l'utilisation de transactions XA, qui empêchait d'éventuels redémarrages d'instances lorsque Enhanced Binlog était activé.
- Correction d'un problème à cause duquel les performances des requêtes se dégradaient lorsque l'optimiseur faisait des estimations de coûts incorrectes en raison de la mise à jour incorrecte des statistiques de l'indice Bufferpool après le redémarrage du serveur de base de données.
- Correction d'un problème qui empêchait les clients de désactiver la fonctionnalité de transfert d'écriture local en raison d'un fil de travail bloqué.
- Correction d'un problème en raison duquel l'exécution de la `SHOW BINARY LOGS` commande prenait plus de temps sur un cluster où Enhanced Binlog était activé ou était activé précédemment. Ce problème pourrait également entraîner une augmentation de la latence de validation si plusieurs `SHOW BINARY LOGS` commandes étaient exécutées simultanément.

Mises à niveau et migrations :

- Correction d'un problème en raison duquel l'application de correctifs ZDP (Zero Downtime Patching) pouvait échouer lors de la tentative de préservation de la connexion d'un utilisateur qui avait été abandonnée. Vous trouverez plus d'informations sur la `DROP USER` commande et son effet sur les connexions actives dans la documentation MySQL.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires, jusqu'à la version 8.0.40 incluse. Pour de plus amples informations, veuillez consulter Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Alors que des transactions importantes étaient reçues et appliquées, et qu'une demande d'arrêt du canal de réplication était faite en utilisant`STOP REPLICA`, MySQL ne l'a pas fait correctement et n'a ensuite traité aucune commande de canal. En outre, le processus d'arrêt du serveur ne s'est pas terminé correctement et a nécessité soit l'arrêt du processus MySQL, soit le redémarrage du système hôte. (Bug #115966, Bug #37008345)

# Mises à jour du moteur de base de données Aurora MySQL 2025-04-07 (version 3.08.2, compatible avec MySQL 8.0.39)

Version : 3.08.2

Aurora MySQL 3.08.2 est généralement disponible. Les versions 3.08 d'Aurora MySQL sont compatibles avec MySQL 8.0.39. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* et 3.08.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.08.2.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

- CVE-2024-11053

Améliorations de la disponibilité :

- Correction d'un problème sur la réplique à cause duquel une interruption du réseau risquait de ne pas rétablir correctement la connexion avec le rédacteur.
- Correction d'un problème en raison duquel une requête à instructions multiples transférée pouvait échouer si elle `innodb_flush_log_at_trx_commit` était définie `0` sur une instance d'écriture.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données Aurora Writer lorsque le transfert d'écriture est activé.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données du lecteur Aurora lors de l'utilisation du transfert d'écriture avec des requêtes de validation implicites ou à instructions multiples.

Améliorations générales

- À partir de cette version d'Aurora MySQL, l'optimisation de l'insertion rapide n'est plus activée. Pour plus d'informations, consultez Améliorations des performances Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Correction d'un problème lié à un dépassement incorrect du `max_user_connections` seuil, qui provoquait des erreurs de connexion pour certains utilisateurs. Cela se produit dans certains cas extrêmes, par exemple lorsque des connexions sont créées et supprimées presque immédiatement.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.39 incluse, en plus des suivantes. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Alors que des transactions importantes étaient reçues et appliquées, et qu'une demande d'arrêt du canal de réplication était faite en utilisant`STOP REPLICA`, MySQL ne l'a pas fait correctement et n'a ensuite traité aucune commande de canal. En outre, le processus d'arrêt du serveur ne s'est pas terminé correctement et a nécessité soit l'arrêt du processus MySQL, soit le redémarrage du système hôte. (Correctifs de bogues communautaires #115966 et #37008345)

## Mises à jour du moteur de base de données Aurora MySQL 2025-02-17 (version 3.08.1, compatible avec MySQL 8.0.39)

Version : 3.08.1

Aurora MySQL 3.08.1 est généralement disponible. Les versions 3.08 d'Aurora MySQL sont compatibles avec MySQL 8.0.39. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* et 3.08.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.08.1.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir

des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

Améliorations de la disponibilité :

- Améliorations apportées qui réduisent le temps de redémarrage de l'instance de base de données lors des redémarrages de bases de données, des mises à niveau des correctifs ou des versions mineures et des basculements sur incident.

Mises à niveau et migrations

- Correction d'un problème qui pouvait provoquer l'échec de Zero Downtime Patching (ZDP) lors de la tentative de restauration des rôles et des autorisations des utilisateurs abandonnés. Pour plus d'informations sur l'`DROP USER`opération, consultez la déclaration DROP USER dans la documentation MySQL.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.39 incluse, en plus des suivantes. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

## Mises à jour du moteur de base de données Aurora MySQL 2024-11-18 (version 3.08.0, compatible avec MySQL 8.0.39)

Version : 3.08.0

Aurora MySQL 3.08.0 est généralement disponible. Les versions 3.08 d'Aurora MySQL sont compatibles avec MySQL 8.0.39. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* et 3.08.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.08.0.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctionnalités

- Ajout de trois nouvelles CloudWatch métriques Amazon pour permettre aux utilisateurs de surveiller le processus de purge d'InnoDB :
  - `PurgeBoundary`
  - `PurgeFinishedPoint`
  - `TruncateFinishedPoint`

Pour plus d'informations, consultez CloudWatchles métriques Amazon pour Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Ajout d'une nouvelle CloudWatch métrique `TransactionAgeMaximum` pour permettre aux utilisateurs d'identifier les transactions de longue durée susceptibles de retarder la limite de purge. Pour plus d'informations, consultez CloudWatch les métriques Amazon pour Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.
- Ajout de trois nouvelles CloudWatch mesures pour éviter out-of-memory (OOM) :
  - `AuroraMillisecondsSpentInOomRecovery`
  - `AuroraNumOomRecoverySuccessful`
  - `AuroraNumOomRecoveryTriggered`

  Pour plus d'informations, consultez CloudWatchles métriques Amazon pour Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.
- Trois CloudWatch indicateurs permettant d'éviter out-of-memory (OOM) ont été modifiés, passant des totaux cumulés à des compteurs incrémentiels :
  - `AuroraMemoryNumDeclinedSqlTotal`
  - `AuroraMemoryNumKillConnTotal`
  - `AuroraMemoryNumKillQueryTotal`

  Pour plus d'informations, consultez CloudWatchles métriques Amazon pour Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.
- Ajout de deux variables d'état globales pour indiquer la quantité de mémoire utilisée par les tables temporaires internes : `aurora_temptable_ram_allocation` et`aurora_temptable_max_ram_allocation`. Ces variables d'état globales augmentent l'observabilité et aident à diagnostiquer les problèmes liés à l'utilisation interne de la mémoire temporaire des tables.

  Pour plus d'informations, consultez les variables d'état globales Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.
- Introduction de la nouvelle variable système `aurora_optimizer_trace_print_before_purge` permettant d'imprimer les traces de l'optimiseur dans le journal des erreurs avant que le serveur ne les purge de la mémoire. Une purge peut être déclenchée en fonction des seuils définis par les variables système optimizer_trace_offset et optimizer_trace_limit.

- Ajout de la prise en charge des classes d'instances de base de données suivantes :
  - db.r7i
  - dbr 8g

  Pour plus d'informations, consultez Moteurs de base de données pris en charge pour les classes d'instance de base de données dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité résolus et CVEs :

- Ajout d'un nouvel utilisateur pour la réplication des journaux binaires (binlog),`rdsrepladmin_priv_checks_user`. Pour plus d'informations, consultez Privilege vérifie l'utilisateur pour la réplication des journaux binaires dans le guide de l'utilisateur Amazon Aurora.
- Correction d'un problème en raison duquel les paramètres d'entrée dans les procédures stockées Aurora MySQL pouvaient être mal gérés.

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 8.0.39. Les correctifs CVE suivants sont inclus :

- CVE-2023-44487
- CVE-2024-0853
- CVE-2024-20996
- CVE-2024-21000
- CVE-2024-21013
- CVE-2024-21125
- CVE-2024-21127
- CVE-2024-21129
- CVE-2024-21130
- CVE-2024-21134
- CVE-2024-21135
- CVE-2024-21137

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

Améliorations de la disponibilité :

- Correction d'un défaut qui pouvait provoquer le redémarrage inattendu du serveur après l'exécution de commandes DML (Data Manipulation Language) sur une table contenant déjà des colonnes supprimées instantanément, de sorte que le nombre de colonnes supprimées et non supprimées était supérieur à 1017.
- Correction d'un problème en raison duquel les fichiers journaux de base de données n'étaient pas correctement pivotés, ce qui entraînait une augmentation de l'utilisation de l'espace de stockage local sur une instance de base de données.
- Correction d'un problème qui pouvait empêcher l'instance de base de données de fermer les connexions lorsque la mémoire était insuffisante lorsque aurora_oom_response était activée, ce qui entraînait un out-of-memory redémarrage qui aurait pu être évité autrement.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur lors de la libération de la mémoire utilisée pour l'application de journalisation.
- Correction d'un problème lié au calcul des métriques internes pour les index de recherche en texte intégral (FTS) susceptible de provoquer le redémarrage de la base de données.

- Correction d'un problème communautaire qui pouvait parfois produire des résultats de requête incorrects lorsqu'une `LIMIT` clause était utilisée dans des sous-requêtes en même temps que le renvoi des conditions d'index vers le bas.
- Correction d'un problème qui provoquait une violation de l'accès à la mémoire entraînant la libération d'un objet mutex n'appartenant plus au thread.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une réplique de journal binaire (binlog) lors du traitement d'un grand nombre de fichiers journaux de relais lors de la restauration du journal de relais.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données du lecteur Aurora lorsque le transfert d'écriture est activé.
- Correction d'un problème en raison duquel une requête contenant plusieurs `UNION` clauses pouvait allouer une grande quantité de mémoire, entraînant le redémarrage d'une instance de base de données.
- Correction d'un problème qui provoquait le redémarrage de l'instance de base de données du rédacteur lors de l'exécution d'une requête parallèle sur une instance de base de données du lecteur.
- Correction d'un problème en raison duquel la réplication du journal binaire se bloquait sur la réplique en raison d'un blocage rencontré sur le I/O thread de la réplique lors de l'exécution de la `FLUSH RELAY LOGS` commande.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données lors de la gestion de grands ensembles de GTID sur un cluster de base de données avec le journal binaire amélioré activé.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une réplique du journal binaire lorsque le cache du journal du relais en mémoire était activé. Le cache de journal de relais en mémoire est activé sur les répliques de journaux binaires gérées par Aurora MySQL lorsque vous utilisez une réplication de journal binaire à thread unique ou une réplication multithread avec le positionnement automatique GTID activé.
- Correction d'un problème qui provoquait le redémarrage d'une instance de réplique binlog lors de l'application d'instructions DCL (Data Control Language) lors du démarrage du moteur de base de données.
- Correction d'un problème qui, dans de rares cas, pouvait provoquer le redémarrage d'une instance de base de données de lecteur en raison d'un blocage lors de l'exécution de `SELECT` requêtes sur des tables mises à jour par son instance de base de données d'écriture.

- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de lecteur de base de données globale Aurora avec une session de transfert d'écriture active.
- Correction d'un problème qui pouvait entraîner le redémarrage des répliques de lecture Aurora en cas de certains rares ordres de validation de transactions sur l'instance de base de données du rédacteur.
- Correction d'un problème qui pouvait entraîner le redémarrage d'une base de données lorsque des événements planifiés étaient annulés alors qu'ils s'exécutaient sur des instances de base de données sur lesquelles le journal binaire amélioré était activé.

Améliorations générales :

- Correction d'un problème à cause duquel une connexion client pouvait se bloquer lors de l'application de correctifs sans interruption (ZDP) ou du redémarrage sans interruption de service (ZDR).
- Correction d'un problème qui, dans de rares cas, provoquait le redémarrage d'une instance de base de données en raison d'un problème de gestion de la mémoire pouvant survenir lors de la gestion d'une défaillance de table ouverte.
- Correction d'un problème qui pouvait provoquer des erreurs inattendues ou des avertissements de violation de clé primaire sur certaines lignes lors de l'exécution d'`INSERT`instructions simultanées sur une table comportant une colonne de clé `AUTO_INCREMENT` primaire et une colonne de clé unique, et lorsqu'une `INSERT` instruction présentait des violations de clé uniques sur différentes lignes.
- L'instrument Performance Schema `memory/sql/sp_head::main_mem_root` est désormais un instrument contrôlé. Par conséquent, la mémoire allouée à l'analyse et à la représentation des programmes stockés contribue désormais aux limites de mémoire par connexion.
- Correction d'un problème en raison duquel une ligne devenait illisible dans l'index spatial lors d'une mise à jour.
- Correction d'un problème qui empêchait les utilisateurs de désactiver le transfert d'écriture local.
- Correction d'un problème à cause duquel une `SELECT COUNT` requête pouvait renvoyer un résultat erroné lors de l'utilisation du transfert d'écriture.
- Correction d'un problème qui pouvait entraîner des résultats de requête incorrects lorsque ZDP restaure de manière incorrecte les variables de session définies comme indices dans les requêtes.
- Correction d'un problème lié à la troncature automatique des tablespaces d'annulation lorsqu'ils sont supérieurs au seuil innodb_max_undo_log_size dans les scénarios de mise à niveau.

- Correction d'un problème en raison duquel la latence de validation et le débit de validation n'étaient pas mesurés lorsque cette `innodb_flush_log_at_trx_commit` valeur est définie sur. `0`
- Correction d'un problème en raison duquel le décalage des répliques d'Aurora était signalé de manière incorrecte.
- Correction d'un problème qui fournissait une valeur incorrecte pour la `threads_running` variable lors de l'utilisation de la base de données globale Aurora.
- Correction d'un problème à cause duquel une réplique du journal binaire Aurora MySQL avec l'optimisation de l'index secondaire parallèle activée redémarrait lors de l'application des modifications de réplication sur des tables contenant des clés étrangères.
- Correction d'un problème de redémarrage causé par le redimensionnement prolongé de la table de hachage verrouillée lors d'événements de redimensionnement ou de réduction.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données après la désactivation de la journalisation binaire, alors que la journalisation binaire améliorée était précédemment activée.
- Des optimisations ont été introduites pour réduire l'utilisation de la mémoire lors de la restauration du dictionnaire de données logique lorsqu'il existe un grand nombre de tables.
- Correction d'un problème à cause duquel un utilisateur pouvait rencontrer une `ERROR 1377 (HY000): Fatal error during log purge` erreur lors de l'exécution de la procédure stockée mysql.rds_set_external_source sur une réplique du journal binaire, alors que la réplication du journal binaire était déjà configurée.
- Correction d'un défaut qui empêchait la persistance des privilèges des rôles utilisateur après la ZDR.
- Correction d'un problème de mémoire associé aux rôles par défaut du définisseur de vue.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors `SHOW VOLUME STATUS` de son exécution.
- Correction d'un problème de redémarrage causé par le redimensionnement prolongé du pool de mémoire tampon lors d'événements de redimensionnement ou de réduction.
- Correction d'un problème de redémarrage causé par l'accès logique à l'avance (LRA) aux pages libérées en raison du redimensionnement de la mémoire tampon lors d'un événement de réduction de la taille.
- Correction d'un problème qui résolvait les `SELECT` requêtes renvoyant des résultats incorrects lorsqu'elles étaient utilisées `LEFT OUTER JOIN` avec une transformation semi-jointive utilisant la matérialisation comme stratégie.

- Correction d'un problème qui pouvait entraîner l'échec du processus de désactivation du transfert d'écriture.
- Correction d'un problème à cause duquel les `BlockedTransactions` CloudWatch indicateurs `ActiveTransactions` et indiquaient des valeurs inférieures aux attentes.
- Correction d'un problème en raison duquel la réplication du journal binaire s'interrompait lorsque la réplique traitait une `DELETE` instruction multitable qui était explicitement supprimée à la fois d'une table parent et d'une table enfant.
- Correction d'un problème qui pouvait entraîner le redémarrage d'une instance de base de données lors du traitement d'un déclencheur avec des fonctions définies par l'utilisateur renvoyant un `enum` type.
- Correction d'un problème en raison duquel la `DMLLatency` CloudWatch métrique affichait des valeurs incorrectes pour une instance de réplique de journal binaire lors de la réplication à l'aide de la commande `binlog_format` set to`row`.
- Correction d'un problème en raison duquel les `UPDATE` requêtes lentes `INSERT` et exécutées par le planificateur d'événements MySQL n'étaient pas enregistrées dans le journal des requêtes lentes sauf si elles étaient précédées d'une requête lente`SELECT`. `DELETE`
- Correction d'un problème qui, dans de rares cas, provoquait soit l'indisponibilité intermittente d'une réplique de lecture Aurora, soit des incohérences dans la définition des tables, parfois liées à l'erreur`Table does not exist`, sur la réplique. Cela est dû à des requêtes de lecture simultanées sur la réplication et aux opérations DDL (Data Definition Language) sur l'instance de base de données du rédacteur.
- Correction d'un problème en raison duquel l'exécution de la `SHOW BINARY LOGS` commande prenait plus de temps sur un cluster de base de données où le journal binaire amélioré était activé ou était activé précédemment. Ce problème pourrait également entraîner une augmentation de la latence de validation si plusieurs `SHOW BINARY LOGS` commandes étaient exécutées simultanément.

Mises à niveau et migrations :

- Amélioration des performances de la mise à niveau des versions majeures d'Aurora MySQL version 2 vers la version 3 pour les clusters de base de données contenant un grand nombre d'objets de base de données (tels que des tables, des déclencheurs et des routines).

Pour les classes d'instances de base de données plus importantes, le processus de mise à niveau de la base de données met à niveau les métadonnées des objets MySQL traditionnels vers le nouveau dictionnaire de données atomique MySQL 8.0 en parallèle à l'aide de plusieurs threads.

- Correction d'un problème qui entraînait l'échec des mises à niveau ou des migrations lorsque la version du moteur de base de données Aurora MySQL cible était la version 3.04.0 ou supérieure. Cela se produit lorsque le paramètre du `lower_case_table_names` cluster de base de données est défini sur et que le classement de base de données MySQL est incompatible avec les noms de table en minuscules. 1

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.39 incluse, en plus des suivantes. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème qui provoquait l'omission incorrecte de `NULL` valeurs dans le jeu de résultats pour certaines requêtes comportant `JOIN` les deux `UNION` opérations. (Correctif de bogue communautaire #114301)

## Mises à jour du moteur de base de données Aurora MySQL 2024-07-23 (version 3.07.1, compatible avec MySQL 8.0.36)

Version : 3.07.1

Aurora MySQL 3.07.1 est généralement disponible. Les versions 3.07 d'Aurora MySQL sont compatibles avec MySQL 8.0.36. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* et 3.07.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.07.1.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité résolus et CVEs :

- Ajout d'un nouvel utilisateur pour la réplication des journaux binaires (binlog),`rdsrepladmin_priv_checks_user`. Pour plus d'informations, consultez Privilege vérifie l'utilisateur pour la réplication des journaux binaires dans le guide de l'utilisateur Amazon Aurora.

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 8.0.36.

Améliorations de la disponibilité :

- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur lors de la libération de la mémoire utilisée pour l'application de journalisation.
- Correction d'un problème lié au calcul des métriques internes pour les index de recherche en texte intégral (FTS) susceptible de provoquer le redémarrage de la base de données.
- Correction d'un problème qui pouvait désactiver la journalisation binaire lorsqu'une erreur se produisait lors de la validation d'une transaction importante.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires, y compris la version 8.0.36. Pour de plus amples informations, veuillez consulter Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

## Mises à jour du moteur de base de données Aurora MySQL 2024-06-04 (version 3.07.0, compatible avec MySQL 8.0.36)

Version : 3.07.0

Aurora MySQL 3.07.0 est généralement disponible. Les versions 3.07 d'Aurora MySQL sont compatibles avec MySQL 8.0.36. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* et 3.07.*.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

Problèmes de sécurité corrigés et CVEs :

- Support activé pour la cryptographie validée par la norme FIPS, une implémentation entièrement détenue. AWS Pour plus d'informations, consultez la section « AWS-LC est désormais certifiée FIPS 140-3 » sur le AWS blog de sécurité.

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 8.0.36. Les correctifs CVE suivants sont inclus :

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

Améliorations de la disponibilité :

- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur lors de la lecture d'une table modifiée ou supprimée sur l'instance de base de données d'écriture.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données Aurora MySQL Writer lorsqu'une session de transfert d'écriture était fermée lors de l'exécution d'une requête transférée.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données lors de la gestion de grands ensembles GTID sur une instance activée pour les journaux binaires.
- Correction d'un problème lors du traitement des `INSERT` requêtes sur les tables partitionnées InnoDB qui pouvait entraîner une diminution progressive de la mémoire libre dans l'instance.
- Correction d'un problème qui, dans de rares cas, pouvait entraîner le redémarrage des instances de base de données du lecteur.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de l'exécution simultanée des instructions SHOW STATUS et PURGE BINARY LOGS. `PURGE BINARY LOGS`est une instruction gérée qui est exécutée pour respecter la période de conservation du journal binaire configurée par l'utilisateur.
- Correction d'un problème qui pouvait provoquer la fermeture inattendue du serveur après l'exécution d'instructions DML (Data Manipulation Language) sur une table dont les colonnes non virtuelles étaient réorganisées avec une `MODIFY COLUMN` instruction or. `CHANGE COLUMN`
- Correction d'un problème qui, lors du redémarrage d'une instance de base de données, pouvait entraîner un redémarrage supplémentaire.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur utilisant le transfert d'écriture lorsqu'une instruction de validation implicite transférée rencontrait une erreur.

- Correction d'un problème qui, dans de rares cas, pouvait provoquer le redémarrage d'une instance de lecteur lors de l'exécution de `SELECT` requêtes sur des tables soumises à une contrainte de clé étrangère.
- Correction d'un problème selon lequel les instances de base de données utilisant des volumes de cluster Aurora de plusieurs To pouvaient subir des temps d'arrêt accrus lors du redémarrage en raison d'échecs de validation du pool de mémoire tampon InnoDB.
- Correction d'un problème qui provoquait le redémarrage d'une base de données lorsqu'une contrainte de clé `DELETE` étrangère `UPDATE` ou en cascade était définie sur une table où une colonne virtuelle était impliquée soit en tant que colonne dans la contrainte de clé étrangère, soit en tant que membre de la table référencée.
- Correction d'un problème qui pouvait interrompre la restauration de la base de données au démarrage si le redémarrage se produisait lors d'opérations d'insertion intensives impliquant des `AUTO_INCREMENT` colonnes.
- Correction d'un problème Aurora Serverless v2 qui pouvait entraîner le redémarrage de la base de données lors de la mise à l'échelle.

Améliorations générales :

- I/O Utilisation réduite et performances améliorées pour un sous-ensemble de requêtes d'analyse par plage de clés primaires utilisant une requête parallèle.
- La version 3.06.0 d'Aurora MySQL a ajouté le support pour l'intégration d'Amazon Bedrock. Dans ce cadre, de nouveaux mots clés réservés (`accept`,`aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint`content_type, ettimeout_ms) ont été ajoutés. Dans Aurora MySQL version 3.07.0, ces mots clés ont été remplacés par des mots clés non réservés, qui sont autorisés comme identifiants sans guillemets. Pour plus d'informations sur la façon dont MySQL gère les mots clés réservés et non réservés, consultez la section Mots clés et mots réservés dans la documentation MySQL.
- Correction d'un problème qui ne renvoyait pas clairement de message d'erreur au client lors de l'appel du service Amazon Bedrock depuis un cluster de base de données Aurora MySQL alors qu' Région AWS Amazon Bedrock n'était pas encore disponible.
- Correction d'un problème qui pouvait entraîner une consommation de mémoire excessive lors de l'interrogation de `BLOB` colonnes à l'aide de la requête parallèle Aurora.
- Ajout de la prise en charge `connection_memory_limit` des `connection_memory_chunk_size` paramètres et à définir au niveau de la session pour qu'ils se comportent de la même manière que dans MySQL Community Edition. Le

`connection_memory_limit` est utilisé pour définir la quantité maximale de mémoire pouvant être utilisée par une connexion utilisateur unique. Le `connection_memory_chunk_size` paramètre peut être utilisé pour définir la taille de segmentation pour les mises à jour du compteur global d'utilisation de la mémoire.

- Correction d'un problème empêchant l'utilisateur d'interrompre une requête ou de définir des délais d'expiration de session pour les `performance_schema` requêtes.
- Problème résolu : la réplication du journal binaire (binlog) configurée pour utiliser des certificats SSL personnalisés (mysql.rds_import_binlog_ssl_material) pouvait échouer lorsque l'instance de réplication était en cours de remplacement d'hôte.
- Ajout de la variable d'état `Aurora_fts_cache_memory_used` globale pour suivre l'utilisation de la mémoire par le système de recherche en texte intégral dans toutes les tables. Pour plus d'informations, consultez les variables d'état globales d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.
- Correction d'un problème selon lequel un cluster Amazon Redshift configuré comme destination zéro ETL pouvait subir une augmentation temporaire IntegrationLaglorsqu'un cluster de base de données Amazon Aurora MySQL était configuré en tant que réplique de journal binaire, avec une intégration Binlog améliorée et Zero-ETL activée.
- Correction d'un problème lié à la gestion des fichiers journaux d'audit qui pouvait rendre les fichiers journaux inaccessibles pour le téléchargement ou la rotation et, dans certains cas, augmenter l'utilisation du processeur.
- Restauration des `AUTO_INCREMENT` clés optimisée pour réduire le temps nécessaire à la restauration des instantanés, à la point-in-time restauration et au clonage de clusters de bases de données contenant un grand nombre de tables dans la base de données.
- Correction d'un problème en raison duquel l'événement wait/io/redo_log_flush n'apparaissait pas dans les tableaux récapitulatifs des événements d'attente du schéma de performance.
- Correction d'un problème qui pouvait provoquer des erreurs clés dupliquées pour les `AUTO_INCREMENT` colonnes utilisant des index décroissants après une restauration instantanée, un retour en arrière ou une opération de clonage de base de données.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données d'écriture lorsqu'une instance de base de données de lecture utilisant le transfert d'écriture exécutait une instruction DML (Data Manipulation Language) contenant une valeur d'horodatage et que le paramètre de `time_zone` base de données était défini sur. `UTC`
- Problème résolu : une `SELECT` requête sur une instance de lecteur Aurora pouvait échouer en raison de l'inexistence de la table d'erreurs lorsque la table contient au moins un index de

recherche en texte intégral (FTS) et qu'une `TRUNCATE` instruction est exécutée sur l'instance de base de données Aurora Writer.

- Correction d'un problème qui, dans de rares cas, entraînait l'échec de l'application de correctifs sans interruption de service (ZDP).
- Correction d'un problème qui pouvait entraîner un ensemble de résultats incomplet lors de l'exécution de requêtes impliquant `LEFT JOIN` ou d'`RIGHT JOIN`opérations utilisant l'algorithme de jointure par hachage avec requête parallèle.

Mises à niveau et migrations :

- Correction d'un problème qui pouvait provoquer des échecs de mise à niveau d'Aurora MySQL version 2 vers Aurora MySQL version 3 lorsqu'une `FTS_DOC_ID` colonne définie par l'utilisateur était présente dans le schéma de table.
- Correction d'un problème qui pouvait provoquer des échecs de mise à niveau d'Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'un problème de synchronisation lors du traitement des tablespaces InnoDB.
- Correction d'un problème qui pouvait entraîner l'échec des mises à niveau majeures vers Aurora MySQL version 3 en raison de la présence d'entrées orphelines pour des espaces disque logiques déjà supprimés dans les tables système InnoDB d'Aurora MySQL version 2.
- Correction d'un problème en raison duquel la valeur SERVER_ID n'était pas mise à jour après un changement de déploiement Amazon RDS. Blue/Green Cela a entraîné des problèmes dans lesquels les pilotes intelligents tels que le pilote JDBC Amazon Web Services (AWS) n'ont pas pu découvrir la topologie du cluster de bases de données après un basculement. blue/green Avec ce correctif, les clusters de base de données Aurora renommés dans le cadre d'un Blue/Green déploiement RDS, qui s'exécutent sur Aurora MySQL version 3.07 ou ultérieure, verront leur `SERVER_ID` valeur mise à jour dans le cadre du passage au numérique. Pour les versions antérieures, les instances de base de données des clusters bleu et vert peuvent être redémarrées pour mettre à jour la `SERVER_ID` valeur.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.36 incluse, en plus des suivantes. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème en raison duquel la valeur de la ligne de cache pouvait être calculée de manière incorrecte, ce qui provoquait un échec lors du redémarrage de la base de données sur les instances basées sur Graviton. (Correctif de bogue communautaire #35479763)
- Correction d'un problème en raison duquel certaines instances de sous-requêtes dans les routines stockées n'étaient pas traitées correctement. (Correctif de bogue communautaire #35377192)
- Correction d'un problème qui pouvait entraîner une augmentation de l'utilisation du processeur en raison de la rotation des certificats TLS en arrière-plan (Community Bug Fix #34284186).
- Correction d'un problème en raison duquel InnoDB autorisait l'ajout de `INSTANT` colonnes aux tables dans le schéma système MySQL dans les versions d'Aurora MySQL inférieures à 3.05, ce qui pouvait entraîner la fermeture inattendue du serveur (redémarrage de l'instance de base de données) après la mise à niveau vers la version 3.05.0 d'Aurora MySQL. (Correctif de bogue communautaire #35625510).

## Mises à jour du moteur de base de données Aurora MySQL 2024-06-26 (version 3.06.1, compatible avec MySQL 8.0.34)

Version : 3.06.1

Aurora MySQL 3.06.1 est généralement disponible. Les versions 3.06 d'Aurora MySQL sont compatibles avec MySQL 8.0.34. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* et 3.07.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise à blue/green niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.06.1.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster de bases de données Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs :

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 8.0.34. Les correctifs CVE suivants sont inclus :

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

Améliorations de la disponibilité :

- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données Aurora MySQL lors de l'exécution d'une requête parallèle.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur lors de la lecture d'une table modifiée ou supprimée sur l'instance de base de données d'écriture.
- Correction d'un problème qui provoquait une violation de l'accès à la mémoire entraînant la libération d'un objet mutex n'appartenant plus au thread.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données Aurora MySQL Writer lorsqu'une session de transfert d'écriture était fermée lors de l'exécution d'une requête transférée.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données lors de la gestion de grands ensembles GTID sur une instance activée pour les journaux binaires.

- Correction d'un problème qui, dans de rares cas, pouvait provoquer le redémarrage d'une instance de lecteur lors de l'exécution de `SELECT` requêtes sur des tables soumises à une contrainte de clé étrangère.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données lors d'une tentative de restauration du dictionnaire de données InnoDB lors de la restauration de la base de données.
- Correction d'un problème Aurora Serverless v2 qui pouvait entraîner le redémarrage de la base de données lors de la mise à l'échelle.

Améliorations générales :

- Correction d'un problème dans le code de publication des métriques en raison duquel la mémoire pouvait être utilisée après avoir été libérée.
- Correction d'un problème qui provoquait des redémarrages répétés du moteur de base de données en raison de l'inexistence d'un objet tablespace d'annulation.
- Correction d'un problème lié à la troncature automatique des tablespaces d'annulation lorsqu'ils sont supérieurs au seuil innodb_max_undo_log_size dans les scénarios de mise à niveau.
- Correction d'un problème qui fournissait une valeur incorrecte pour la variable d'`threads_running`état lors de l'utilisation de la base de données globale Aurora.
- Correction d'un problème à cause duquel une réplique de lecture du journal binaire Aurora MySQL (binlog) avec optimisation de l'index secondaire parallèle activée redémarrait lors de l'application de modifications de réplication sur des tables contenant des clés étrangères.
- La version 3.06.0 d'Aurora MySQL a ajouté le support pour l'intégration d'Amazon Bedrock. Dans ce cadre, de nouveaux mots clés réservés (`accept`,`aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint`, `content_type`, et `timeout_ms`) ont été ajoutés. Dans la version 3.06.1 d'Aurora MySQL, ces mots clés ont été remplacés par des mots clés non réservés, autorisés comme identifiants sans guillemets. Pour plus d'informations sur la façon dont MySQL gère les mots clés réservés et non réservés, consultez la section Mots clés et mots réservés dans la documentation MySQL.
- Correction d'un problème qui ne renvoyait pas clairement de message d'erreur au client lors de l'appel du service Amazon Bedrock depuis un cluster de base de données Aurora MySQL alors qu' Région AWS Amazon Bedrock n'était pas encore disponible.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données en raison d'informations inexactes sur le détenteur du verrou `rw_lock` lors de l'utilisation de lectures parallèles.

- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors `SHOW VOLUME STATUS` de son exécution.
- Correction d'un problème de gestion de la mémoire qui entraînait une diminution de la mémoire disponible au fil du temps lors de l'exécution de `SELECT ... INTO OUTFILE ...` requêtes.
- Ajout de la prise en charge des `connection_memory_chunk_size` paramètres `connection_memory_limit` et à définir au niveau de la session pour qu'ils se comportent de la même manière que les fonctionnalités correspondantes dans MySQL Community Edition. Le `connection_memory_limit` paramètre définit la quantité maximale de mémoire pouvant être utilisée par une connexion utilisateur unique. Le `connection_memory_chunk_size` paramètre définit la taille de segmentation pour les mises à jour du compteur global d'utilisation de la mémoire.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lorsque le stockage local de l'instance de base de données atteignait sa pleine capacité.
- Correction d'un problème en raison duquel le schéma de performance n'était pas activé lorsque la gestion automatisée de Performance Insights était activée pour les instances de base de données db.t4g.medium et db.t4g.large.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données d'écriture lorsqu'une instance de base de données de lecture utilisant le transfert d'écriture exécutait une instruction DML (Data Manipulation Language) contenant une valeur d'horodatage et que le paramètre de `time_zone` base de données était défini sur. `UTC`
- Correction d'un problème lié à l'application de correctifs sans interruption de service (ZDP) qui empêchait une instance de base de données de fermer les connexions client lorsqu'elle atteignait la valeur minimale configurée par le client de l'un ou de l'autre. `wait_timeout` `interactive_timeout`

Mises à niveau et migrations :

- Correction d'un problème qui entraînait l'échec des mises à niveau ou des migrations lorsque la version du moteur de base de données Aurora MySQL cible était la version 3.04.0 ou supérieure. Cela se produit lorsque le paramètre du `lower_case_table_names` cluster de base de données est défini sur et que le classement de base de données MySQL est incompatible avec les noms de table en minuscules. 1

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires, y compris la version 8.0.34. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

## Mises à jour du moteur de base de données Aurora MySQL 2024-03-07 (version 3.06.0, compatible avec MySQL 8.0.34)

Version : 3.06.0

Aurora MySQL 3.06.0 est généralement disponible. Les versions 3.06 d'Aurora MySQL sont compatibles avec MySQL 8.0.34. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* et 3.06.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise à blue/green niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.06.0.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster de bases de données Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctionnalités

- Aurora MySQL version 3.06.0 prend en charge l'intégration d'Amazon Bedrock et introduit les nouveaux mots clés réservés`accept`,, `aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint`, `content_type` et. `timeout_ms` Vérifiez les définitions des objets pour vérifier l'utilisation des nouveaux mots clés réservés avant de passer à la version 3.06.0. Pour atténuer le conflit avec les nouveaux mots clés réservés, citez les mots clés réservés utilisés dans les définitions des objets. Pour plus d'informations sur l'intégration d'Amazon Bedrock et la gestion des mots clés réservés, consultez Qu'est-ce qu'Amazon Bedrock ? dans le guide de l'utilisateur d'Amazon Aurora. Pour plus d'informations, consultez les sections Mots clés et mots réservés, la table INFORMATION_SCHEMA KEYWORDS et les noms d'objets de schéma dans la documentation MySQL.
- Performances améliorées pour les répliques de journaux binaires lors de la réplication de transactions pour de grandes tables comportant plusieurs index secondaires. Cette fonctionnalité introduit un pool de threads permettant d'appliquer des modifications d'index secondaires en parallèle sur une réplique binlog. La fonctionnalité est contrôlée par le paramètre de `aurora_binlog_replication_sec_index_parallel_workers` cluster de base de données, qui contrôle le nombre total de threads parallèles disponibles pour appliquer les modifications d'index secondaires. Pour plus d'informations, consultez Réplication avec Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Ajout d'une nouvelle procédure stockée `mysql.rds_set_read_only` qui permet de modifier la valeur de la variable système globale `read_only` sur les instances de base de données de votre cluster Aurora MySQL. Pour plus d'informations, consultez la section Réplication dans le guide de l'utilisateur Amazon Aurora.
- Ajout d'une nouvelle procédure stockée `mysql.rds_set_binlog_source_ssl` qui permet de définir le chiffrement d'une réplique de journal binaire en spécifiant une valeur pour`SOURCE_SSL`. Pour plus d'informations, consultez la section Réplication dans le guide de l'utilisateur Amazon Aurora.
- Amazon Aurora Machine Learning est une intégration optimisée entre la base de données Aurora MySQL et les services d'apprentissage AWS automatique (ML). Amazon Bedrock est désormais pris en charge, ce qui vous permet d'invoquer des modèles d'apprentissage automatique dans

Amazon Bedrock directement depuis votre cluster de bases de données Aurora MySQL à l'aide de SQL. Pour plus d'informations sur l'utilisation d'Amazon Bedrock avec votre cluster de bases de données Aurora MySQL, consultez la section Utilisation de l'apprentissage automatique Amazon Aurora avec Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

- Aurora MySQL version 3.06 ajoute la prise en charge de l'annulation automatique de la troncature des tablespaces. Cette optimisation vous permet de récupérer l'espace inutilisé dans les tablespaces d'annulation une fois les journaux d'annulation purgés.

## Améliorations

Problèmes de sécurité corrigés et CVEs :

Les correctifs CVE suivants sont inclus dans cette version :

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

Améliorations de la disponibilité :

- Correction d'un problème en raison duquel une instance de base de données de réplication en lecture ne pouvait pas être lancée correctement lorsque la charge de travail de l'instance de base de données d'écriture était élevée.
- Correction d'un problème à cause duquel une instance de base de données Aurora MySQL Writer pouvait basculer en raison d'un défaut de communication avec le stockage Aurora. Le défaut est dû à une rupture de communication entre l'instance de base de données et le stockage sous-jacent suite à une mise à jour logicielle de l'instance de stockage Aurora.
- Correction d'un problème lors du traitement des `INSERT` requêtes sur les tables partitionnées InnoDB qui pouvait entraîner une diminution progressive de la mémoire libre dans l'instance.
- Correction d'un problème qui pouvait provoquer le redémarrage ou le basculement d'une instance de base de données Aurora MySQL en raison d'une diminution de la mémoire disponible lorsque la jointure par hachage est utilisée lors de l'exécution de requêtes.

- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de l'exécution simultanée des instructions SHOW STATUS et PURGE BINARY LOGS. `PURGE BINARY LOG`Sest une instruction gérée qui est exécutée pour respecter la période de conservation du journal binaire configurée par l'utilisateur.
- Correction d'un problème qui pouvait provoquer la fermeture inattendue du serveur après l'exécution d'instructions DML (Data Manipulation Language) sur une table dont les colonnes non virtuelles étaient réorganisées avec une `MODIFY COLUMN` instruction or. `CHANGE COLUMN`
- Correction d'un problème qui, lors du redémarrage d'une instance de base de données, pouvait entraîner un redémarrage supplémentaire.
- Correction d'un problème qui provoquait le redémarrage de la base de données lorsqu'une contrainte de clé `DELETE` étrangère `UPDATE` ou en cascade était définie sur une table où une colonne virtuelle était impliquée soit en tant que colonne dans la contrainte de clé étrangère, soit en tant que membre de la table référencée.
- Dans Aurora MySQL 2.10, nous avons ajouté la prise en charge du redémarrage d'un cluster de base de données Aurora avec disponibilité en lecture. Cette fonctionnalité permet aux instances de base de données de lecteur de rester en ligne pendant le redémarrage d'une instance de base de données d'écriture. Cette fonctionnalité est désormais prise en charge sur le secondaire Régions AWS dans les bases de données globales Aurora MySQL, ce qui garantit que vous pouvez toujours répondre aux demandes de lecture lors du redémarrage d'une instance d'écriture sur le cluster principal. Auparavant, lors du redémarrage d'une instance d'écriture, toutes les instances de lecteur d'un cluster secondaire Aurora MySQL redémarraient également. Avec cette version, les instances de lecteur de cluster secondaires continuent de traiter les demandes de lecture lors du redémarrage d'une instance d'écriture, ce qui améliore la disponibilité de lecture dans le cluster. Pour plus d'informations, voir Redémarrage d'un cluster Aurora avec disponibilité en lecture.
- Correction d'un problème qui pouvait interrompre la restauration de la base de données au démarrage si le redémarrage se produisait lors d'opérations d'insertion intensives impliquant des `AUTO_INCREMENT` colonnes.

Améliorations générales :

- Correction d'un problème qui pouvait entraîner l'échec d'une requête parallel en raison de problèmes réseau transitoires lors de la lecture des données du volume du cluster Aurora.
- Correction d'un problème empêchant l'utilisateur d'interrompre une requête ou de définir des délais d'expiration de session pour les `performance_schema` requêtes.

- Problème résolu : la réplication du journal binaire (binlog) configurée pour utiliser des certificats SSL personnalisés (mysql.rds_import_binlog_ssl_material) pouvait échouer lorsque l'instance de réplication était en cours de remplacement d'hôte.
- Les petites instances de base de données dont la mémoire est inférieure ou égale à 4 GiB ferment désormais les connexions les plus gourmandes en mémoire lorsque l'instance de base de données est soumise à une pression de mémoire. Vous pouvez également ajuster le pool de mémoire tampon pour réduire sa taille. Pour plus d'informations, consultez la section out-of-memoryProblèmes liés à Amazon Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.
- Modification de la réponse par défaut pour`aurora_oom_response`, sur toutes les classes d'instance de base de données disposant de plus de 4 GiB de mémoire, de vide à. `print` Pour plus d'informations, consultez la section out-of-memoryProblèmes liés à Amazon Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.
- Correction d'un problème lié à la gestion des fichiers journaux d'audit qui pouvait rendre les fichiers journaux inaccessibles pour le téléchargement ou la rotation et, dans certains cas, augmenter l'utilisation du processeur.
- Restauration des `AUTO_INCREMENT` clés optimisée pour réduire le temps nécessaire à la restauration des instantanés, à la point-in-time restauration et au clonage de clusters de bases de données contenant un grand nombre de tables dans la base de données.
- Correction d'un problème en raison duquel l'événement wait/io/redo_log_flush n'apparaissait pas dans les tableaux récapitulatifs des événements d'attente du schéma de performance.
- Ajout des `Aurora_lockmgr_buffer_pool_memory_used` métriques `Aurora_lockmgr_memory_used` et pour suivre l'utilisation de la mémoire par le gestionnaire de verrous. Pour plus d'informations, consultez les variables d'état globales Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.
- Correction d'un problème à cause duquel les petites instances de répliques en lecture pouvaient subir un délai de réplication accru après une mise à niveau à partir de versions d'Aurora MySQL inférieures à 2.11.*.
- Correction d'un problème qui pouvait provoquer des erreurs clés dupliquées pour les `AUTO_INCREMENT` colonnes utilisant des index décroissants après une restauration instantanée, un retour en arrière ou une opération de clonage de base de données.
- Problème résolu : une `SELECT` requête sur une instance de lecteur Aurora pouvait échouer en raison de l'inexistence de la table d'erreurs lorsque la table contient au moins un index de recherche en texte intégral (FTS) et qu'une `TRUNCATE` instruction est exécutée sur l'instance de base de données Aurora Writer.

- Correction d'un problème qui pouvait entraîner un ensemble de résultats incomplet lors de l'exécution de requêtes impliquant `LEFT JOIN` ou d'`RIGHT JOIN`opérations utilisant l'algorithme de jointure par hachage avec requête parallèle.

Mises à niveau et migrations :

- Correction d'un problème qui pouvait entraîner l'échec des mises à niveau des versions majeures en cas de présence d'une `FTS_DOC_ID` colonne définie par l'utilisateur dans le schéma de table.
- Correction d'un problème qui pouvait provoquer des échecs de mise à niveau d'Aurora My SQLversion 2 vers Aurora MySQL version 3 en raison d'un problème de synchronisation lors du traitement des tablespaces InnoDB.
- Correction d'un problème qui pouvait entraîner l'échec des mises à niveau majeures vers Aurora MySQL version 3 en raison de la présence d'entrées orphelines pour des espaces disque logiques déjà supprimés dans les tables système InnoDB d'Aurora MySQL version 2.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.34 incluse, en plus des suivantes. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème en raison duquel la valeur de la ligne de cache pouvait être calculée de manière incorrecte, ce qui provoquait un échec lors du redémarrage de la base de données sur les instances basées sur Graviton. (Correctif de bogue communautaire #35479763)
- Correction d'un problème en raison duquel certaines instances de sous-requêtes dans les routines stockées n'étaient pas toujours traitées correctement. (Correctif de bogue communautaire #35377192)
- Correction d'un problème qui pouvait entraîner une augmentation de l'utilisation du processeur en raison de la rotation des certificats TLS en arrière-plan (Community Bug Fix #34284186).
- Correction d'un problème en raison duquel InnoDB autorisait l'ajout de `INSTANT` colonnes aux tables dans le schéma système MySQL dans les versions d'Aurora MySQL inférieures à 3.05, ce qui pouvait entraîner la fermeture inattendue du serveur (redémarrage de l'instance de base de données) après la mise à niveau vers la version 3.05.0 d'Aurora MySQL. (Correctif de bogue communautaire #35625510).

# Mises à jour du moteur de base de données Aurora MySQL 2024-01-31 (version 3.05.2, compatible avec MySQL 8.0.32)

Version : 3.05.2

Aurora MySQL 3.05.2 est généralement disponible. Les versions 3.05 d'Aurora MySQL sont compatibles avec MySQL 8.0.32. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.* et 3.05.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.05.2.

Pour plus d'informations sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau pour Aurora MySQL version 3. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes de mise à niveau avec Aurora MySQL version 3 dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs :

Les correctifs CVE suivants sont inclus dans cette version :

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

Améliorations de la disponibilité :

- Correction d'un problème en raison duquel le traitement des `INSERT` requêtes sur les tables partitionnées d'InnoDB pouvait entraîner une diminution progressive de la mémoire libre dans l'instance.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de l'exécution simultanée des instructions SHOW STATUS et PURGE BINARY LOGS. `PURGE BINARY LOGS`est une instruction gérée qui est exécutée pour respecter la période de conservation du journal binaire configurée par l'utilisateur.
- Correction d'un problème qui pouvait provoquer la fermeture inattendue du serveur après l'exécution d'instructions DML (Data Manipulation Language) sur une table dont les colonnes non virtuelles étaient réorganisées avec une `MODIFY COLUMN` instruction or. `CHANGE COLUMN`
- Correction d'un problème qui, lors du redémarrage d'une instance de base de données, pouvait entraîner un redémarrage supplémentaire.

Améliorations générales :

- Correction d'un problème empêchant l'utilisateur d'interrompre une requête ou de définir des délais d'expiration de session pour les `performance_schema` requêtes.
- Correction d'un problème en raison duquel la configuration de la réplication du journal binaire (binlog) à l'aide de certificats SSL personnalisés (mysql.rds_import_binlog_ssl_material) pouvait échouer lorsque l'instance de réplication était en cours de remplacement d'hôte.
- Correction d'un problème lié à la gestion des fichiers journaux d'audit qui pouvait rendre les fichiers journaux inaccessibles pour le téléchargement ou la rotation et, dans certains cas, augmenter l'utilisation du processeur.
- Dans les versions d'Aurora MySQL inférieures à la version 3.05.2, les utilisateurs ne peuvent pas récupérer la sortie des instances de `SHOW ENGINE INNODB STATUS` base de données du lecteur Aurora MySQL. Cela est dû au comportement par défaut d'InnoDB lorsqu'il `innodb_read_only` est activé.

Dans Aurora MySQL version 3.05.2 et versions supérieures, lorsqu'il `SHOW ENGINE INNODB STATUS` est exécuté sur une instance de lecteur, le résultat est écrit dans le journal des erreurs MySQL, ce qui facilite le dépannage.

Pour plus d'informations sur l'utilisation des journaux d'erreurs MySQL, consultez la section Journaux d'erreurs MySQL d'Aurora. Pour plus d'informations`SHOW ENGINE INNODB STATUS`, consultez l'instruction SHOW ENGINE dans la documentation MySQL.

Mises à niveau et migrations :

- Correction d'un problème qui pouvait provoquer des échecs de mise à niveau d'Aurora MySQL version 2 vers Aurora MySQL version 3 lorsqu'une `FTS_DOC_ID` colonne définie par l'utilisateur était présente dans le schéma de table.
- Correction d'un problème qui pouvait provoquer des échecs de mise à niveau d'Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'un problème de synchronisation lors du traitement des tablespaces InnoDB.
- Correction d'un problème qui pouvait entraîner l'échec des mises à niveau majeures vers la version 3 d'Aurora MySQL en raison de la présence d'entrées orphelines pour des espaces disque logiques déjà supprimés dans les tables système InnoDB d'Aurora MySQL version 2.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.32 incluse, en plus des suivantes. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème en raison duquel un nombre excessif de lectures de disque étaient `records_in_range` effectuées pour les `INSERT` opérations, ce qui entraînait une baisse progressive des performances. (Correctif de bogue communautaire #34976138)

## Mises à jour du moteur de base de données Aurora MySQL 21/11/2023 (version 3.05.1, compatible avec MySQL 8.0.32)

Version : 3.05.1

Aurora MySQL 3.05.1 est généralement disponible. Les versions 3.05 d'Aurora MySQL sont compatibles avec MySQL 8.0.32. Pour plus d'informations, consultez les notes de mise à jour de MySQL 8.0.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* et 3.05.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 3.* existant vers Aurora MySQL 3.05.1. Vous pouvez également restaurer un instantané à partir de n'importe quelle version d'Aurora MySQL actuellement prise en charge dans Aurora MySQL 3.05.1.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 3.05.*, vous devez mettre à niveau vos clusters de bases de données principales et secondaires vers la même version, y compris le même niveau de correctif. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Cette version inclut tous les CVEs correctifs communautaires, y compris MySQL 8.0.32.

- CVE-2023-38545

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.32 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème dans InnoDB : si une `INSTANT ADD` colonne était ajoutée à une table MySQL dans un schéma système entre les versions 3.01 à Aurora MySQL 3.04 et après la mise à niveau d'Aurora MySQL vers la version 3.05.0, le serveur se fermait de manière inattendue DMLs sur ces tables. (Correctif de bogue Community n° 35625510)

# Mises à jour du moteur de base de données Aurora MySQL du 30/10/2023 (version 3.05.0.1, compatible avec MySQL 8.0.32), Bêta

Version : 3.05.0.1

Aurora MySQL 3.05.0.1 est généralement disponible dans les régions suivantes : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Californie du Nord), USA Ouest (Oregon), (US-Est) et AWS GovCloud AWS GovCloud (US-Ouest). Il s'agit d'une version anticipée qui ne contient que des correctifs de sécurité. Ces correctifs seront déployés plus largement dans toutes les régions avec le prochain correctif (version 3.05.1). Les versions 3.05 d'Aurora MySQL sont compatibles avec MySQL 8.0.32. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* et 3.05.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 3.* vers Aurora MySQL 3.05.0.1. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 3.05.0.1.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 3.05.*, vous devez mettre à niveau vos clusters de bases de données principales et secondaires vers la même version, y compris le même niveau de correctif. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Cette version inclut tous les CVEs correctifs communautaires, y compris MySQL 8.0.32.

- CVE-2023-38545

# Mises à jour du moteur de base de données Aurora MySQL du 25/10/2023 (version 3.05.0, compatible avec MySQL 8.0.32)

Version : 3.05.0

Aurora MySQL 3.05.0 est disponible. Les versions 3.05 d'Aurora MySQL sont compatibles avec MySQL 8.0.32. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.* et 3.05.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.05.0.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Ajout de la prise en charge de l'enregistrement des données d'un cluster de base de données Aurora MySQL dans des fichiers texte stockés dans un compartiment Amazon S3 chiffré avec une clé KMS (SSE-KMS). Pour plus d'informations, consultez Enregistrement de données d'un cluster de bases de données Amazon Aurora MySQL dans des fichiers texte stockés dans un compartiment Amazon S3.
- Introduction d'une nouvelle variable d'état globale `aurora_tmz_version` pour indiquer la version actuelle des informations de fuseau horaire utilisées par le moteur. Les valeurs suivent la version de la base de données de fuseaux horaires de l'IANA et sont formatées comme YYYYsuffix « », par exemple 2022a et 2023c. Pour plus d'informations, consultez Variables d'état globales Aurora MySQL.

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs de CVE supplémentaires ci-dessous :

- CVE-2022-37434

Améliorations de la disponibilité :

- Correction d'un problème de redémarrage des instances de base de données Aurora MySQL utilisant une requête parallèle lorsqu'elles exécutent un grand nombre de requêtes parallèles simultanées.
- Correction d'un problème de conflit de verrouillage causé par un thread de journalisation d'audit pouvant conduire à une utilisation élevée du processeur et à des délais d'attente pour les applications clientes.
- Correction d'un problème pouvant conduire à la récupération incorrecte du jeu de GTID exécuté sur un cluster de réplica du journal binaire avec le journal binaire amélioré activé lorsqu'une source binlog est définie sur `gtid_mode` `ON` ou `ON_PERMISSIVE`. Ce problème peut entraîner un redémarrage supplémentaire de l'instance d'enregistreur du cluster de réplica pendant la récupération, ou entraîner des résultats incorrects lors de l'interrogation du jeu de GTID exécuté.
- Correction d'un problème de gestion de la mémoire pouvant entraîner le redémarrage ou le basculement d'une instance de base de données Aurora MySQL en raison d'une diminution de la mémoire disponible lorsque le journal binaire amélioré est activé.

- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données lors d'une tentative de lecture d'une page de base de données appartenant à une table supprimée.
- Correction d'un problème susceptible d'entraîner le redémarrage de l'instance d'enregistreur lorsque celle-ci augmente le volume de la base de données jusqu'à un multiple de 160 Go.
- Correction d'un problème de blocage d'une instance de base de données MySQL Aurora avec la fonctionnalité de journal binaire amélioré activée pendant le démarrage de l'instance de base de données alors que le processus de récupération du journal binaire est en cours d'exécution.
- Correction d'un problème de redémarrages multiples d'une instance de base de données Aurora MySQL pendant le démarrage de l'instance alors que de grands segments d'annulation sont initialisés.
- Correction d'un problème d'application de correctifs sans temps d'arrêt, qui provoque le redémarrage de l'instance et la fermeture inattendue des connexions à la base de données.
- Correction d'un problème susceptible de provoquer le redémarrage d'une instance de base de données en raison d'un blocage lors de l'exécution simultanée des instructions SHOW STATUS et PURGE BINARY LOGS. L'instruction gérée purge binary logs est exécutée pour respecter la durée de conservation du journal binaire configurée par l'utilisateur.
- Correction d'un problème susceptible d'entraîner l'indisponibilité du cluster de base de données si l'instance d'enregistreur redémarre alors que la base de données crée ou supprime des déclencheurs sur des tables système internes.
- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données en raison d'un long délai d'attente du sémaphore lors de l'utilisation de la fonctionnalité de journal binaire amélioré sur un cluster doté d'un réplica Aurora.
- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données lors de l'exécution d'une requête référençant une fonction d'agrégation.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage d'une instance de base de données lorsqu'Aurora Serverless v2 tente de manière incorrecte de mettre à jour le cache de la table lors de la mise à l'échelle.
- Correction d'un problème de prise en compte de méthodes d'accès à l'analyse d'index non prises en charge pour les expressions de table courantes (CTE) lors de la matérialisation des tables temporaires intermédiaires, ce qui pouvait entraîner des comportements indésirables, notamment des redémarrages de la base de données ou des résultats de requête incorrects. Nous avons résolu ce problème en évitant d'utiliser de telles méthodes d'accès non prises en charge par analyse d'index sur les tables utilisant le moteur TempTable de stockage.

Améliorations générales :

- Correction d'un problème susceptible d'entraîner l'indisponibilité de la base de données lorsque le journal binaire amélioré est activé sur un cluster de base de données Aurora Serverless v2 exécuté sur Aurora MySQL 3.04.0.
- Suppression des métadonnées de stockage inutilisées avant d'écrire sur le stockage Aurora lorsque la fonctionnalité de journal binaire amélioré est activée. Cette suppression permet d'éviter certains scénarios de redémarrage ou de basculement de base de données en raison d'une latence d'écriture accrue due à l'augmentation du nombre d'octets transmis sur le réseau.
- Avec l'ajout des tables `malloc_stats` et `malloc_stats_totals` dans le paramètre `performance_schema`, trois variables système avancées ont été ajoutées pour contrôler le comportement de Jemalloc, un allocateur de mémoire interne :
  - `aurora_jemalloc_background_thread`.
  - `aurora_jemalloc_dirty_decay_ms`.
  - `aurora_jemalloc_tcache_enabled`.
- Correction d'un problème d'absence de création de tables de schéma de performance spécifiques à Aurora lors d'une mise à niveau ou d'une migration.
- Ajout d'une nouvelle variable système, `aurora_use_vector_instructions`. Lorsque ce paramètre est activé, Aurora MySQL utilise des instructions de traitement vectoriel optimisées pour améliorer les performances lors de I/O lourdes charges de travail. Ce paramètre est activé par défaut dans Aurora MySQL versions 3.05 et ultérieures. Pour plus d'informations, consultez Paramètres de configuration d'Aurora MySQL.
- Correction d'un problème en raison duquel les `NumBinaryLogFiles` métriques affichées pouvaient CloudWatch afficher des résultats incorrects lorsque le journal binaire amélioré était activé.
- Le délai d'attente des requêtes pour les opérations de machine learning Aurora MySQL adressées à Amazon SageMaker est passé de 3 à 30 secondes. Cette augmentation permet de résoudre le problème qui amenait les clients à constater une augmentation du nombre de tentatives ou d'échecs des requêtes adressées à Amazon SageMaker par le machine learning Aurora MySQL lors de l'utilisation de lots plus volumineux.
- Ajout de la prise en charge des tables `malloc_stats` et `malloc_stats_totals` dans la base de données performance_schema.
- Mise à jour du mot-clé `FROM` de la commande `LOAD DATA FROM S3` afin de le rendre facultatif. Pour plus d'informations, consultez Chargement de données dans un cluster de bases de données Amazon Aurora MySQL à partir de fichiers texte stockés dans un compartiment Amazon S3.

- Ajout de la prise en charge du paramètre `innodb_aurora_instant_alter_column_allowed`, qui contrôle si l'algorithme `INSTANT`peut être utilisé pour les opérations `ALTER COLUMN`. Pour plus d'informations, consultez Paramètres de niveau cluster.
- Correction d'un problème empêchant l'établissement de nouvelles connexions clientes à la base de données lorsque le transfert d'écriture est activé.
- Correction d'un problème empêchant la modification du paramètre `table_open_cache` de la base de données de prendre effet tant que l'instance de base de données n'est pas redémarrée.
- Correction d'un problème susceptible de provoquer des erreurs de duplication de clés dans les colonnes `AUTO_INCREMENT` utilisant des index décroissants après une restauration instantanée, un retour en arrière ou une opération de clonage de base de données.
- Correction d'un problème avec l'analyse d'index qui pouvait renvoyer un résultat inexact lors de l'exécution d'une requête `SELECT` avec la clause `GROUP BY` et le paramètre `aurora_parallel_query` défini sur `ON`.
- Correction d'un problème susceptible d'épuiser la mémoire disponible lors de l'exécution de requêtes sur la table `INFORMATION_SCHEMA INNODB_TABLESPACES`.
- Correction d'un problème empêchant l'instance de lecteur d'ouvrir une table (ERROR 1146). Ce problème se produit lors de l'exécution de certains types d'instructions DDL (Data Definition Language) en ligne alors que l'algorithme `INPLACE` est utilisé sur l'instance d'enregistreur.
- Correction d'un problème empêchant le redémarrage de l'instance pendant la mise à l'échelle d'Aurora Serverless v2 lorsque le processus de surveillance interne soumet par erreur des demandes de mise à l'échelle en double.
- Correction d'un problème qui pouvait provoquer le redémarrage de la base de données lorsque les consommateurs de journaux binaires (binlog) connectés utilisaient un serveur de réplication de journaux binaires dupliqué. IDs
- Introduction d'un cache de journal de relais en mémoire pour les réplicas de journal binaire gérés par Aurora MySQL. Grâce à cette amélioration, le débit de réplication de journal binaire peut être accru de 40 %. Cette amélioration est activée automatiquement lorsque la réplication de journal binaire monothread est utilisée ou que la réplication multithread est utilisée avec le positionnement automatique GTID activé.

Mises à niveau et migrations :

- Lors de la mise à niveau de MySQL 5.7 vers MySQL 8.0 avec un très grand nombre de tables dans une seule base de données, le serveur consommait trop de mémoire. Lors du processus de vérification de la possibilité de mise à niveau des tables, il a été constaté que nous récupérions

tous les objets `Table` du dictionnaire de données en amont et que nous les traitions, récupérions leur nom, et vérifions la compatibilité des versions dans la liste. La recherche préalable de tous les objets n'était pas nécessaire dans ce cas, et contribuait grandement à la consommation de mémoire. Pour corriger ce problème, nous récupérons désormais un objet `Table` à la fois, nous effectuons toutes les vérifications nécessaires, nous récupérons son nom et nous libérons l'objet avant de passer au suivant. (Bogue n° 34526001)

- Amélioration des performances des mises à niveau des versions majeures d'Aurora MySQL de la version 2 à la version 3 en exécutant des vérifications de tablespace en parallèle en utilisant tous les v disponibles CPUs sur l'instance de base de données.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.32 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème susceptible d'augmenter l'utilisation du processeur en raison de la rotation des certificats TLS en arrière-plan. (Correctif du bogue n° 34284186)

## Mises à jour du moteur de base de données Aurora MySQL 2025-05-05 (version 3.04.4, compatible avec MySQL 8.0.28)

Version : 3.04.4

Aurora MySQL 3.04.4 est généralement disponible. Les versions 3.04 d'Aurora MySQL sont compatibles avec MySQL 8.0.28. Pour plus d'informations sur les modifications apportées, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour plus d'informations sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, consultez Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition dans le guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
> Nous vous recommandons de ne pas définir le `AutoMinorVersionUpgrade` paramètre sur `true` (ou de ne pas activer la mise à niveau automatique des versions mineures dans AWS Management Console) pour les versions LTS. Cela pourrait entraîner la mise à niveau de votre cluster de base de données vers la prochaine version cible pour la campagne de mise à niveau automatique des versions mineures, qui n'est peut-être pas une version LTS.

Vous pouvez effectuer une mise à niveau sur place qui exploite un zero-downtime-patchcluster Aurora MySQL version 3.04.4, en restaurer un instantané ou lancer une blue/green mise à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité

Critique CVEs :

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

Améliorations de disponibilité

- Correction d'un problème sur la réplique à cause duquel une interruption du réseau risquait de ne pas rétablir correctement la connexion avec le rédacteur.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une réplique de journal binaire (binlog) lors du traitement d'un grand nombre de fichiers journaux de relais lors de la restauration du journal de relais.
- Correction d'un problème qui provoquait le redémarrage d'une instance de lecteur de base de données lors de l'exécution d'une requête à l'aide de la fonctionnalité Parallel Query.
- Correction d'un problème qui, dans de rares cas, pouvait désactiver la journalisation binaire lorsqu'une erreur se produisait lors de la validation d'une transaction importante.
- Correction d'un problème qui pouvait entraîner le redémarrage des répliques de lecture Aurora en cas de certains rares ordres de validation de transactions sur l'instance de base de données du rédacteur.
- Correction d'un problème qui pouvait entraîner le redémarrage de la base de données lorsque des événements planifiés étaient annulés pendant l'exécution sur des instances sur lesquelles Enhanced Binlog était activé.
- Correction d'un problème selon lequel les instances de base de données utilisant des volumes de cluster Aurora de plusieurs To pouvaient subir des temps d'arrêt accrus au redémarrage en raison d'échecs de validation du pool de mémoire tampon InnoDB.

Améliorations générales

- Les privilèges suivants ont été ajoutés aux `rds_superuser_role` : `FLUSH_OPTIMIZER_COSTS`FLUSH_STATUS,`FLUSH_TABLES`, et`FLUSH_USER_RESOURCES`. Pour plus d'informations à ce sujet`rds_superuser_role`, consultez la section Amazon Master User Accounts with Amazon Aurora dans le guide de l'utilisateur Amazon Aurora. Pour plus d'informations sur ces privilèges dynamiques, consultez la documentation MySQL.
- Correction d'un problème en raison duquel une connexion préservée était mal gérée lors de l'application de correctifs sans interruption (ZDP) ou du redémarrage sans interruption (ZDR), ce qui pouvait amener le client à attendre indéfiniment la fin d'une requête.
- Correction d'un problème en raison duquel la ligne devenait illisible via l'index spatial lors d'une mise à jour.
- Correction d'un problème en raison duquel une requête contenant un indice d'optimisation abandonnée lors d'un redémarrage sans interruption ou d'une opération de correction sans interruption de service pouvait être traitée de manière incorrecte.

- Correction d'un problème en raison duquel la latence de validation n'était pas mesurée lorsqu'elle `innodb_flush_log_at_trx_commit` était définie sur`0`.
- Des optimisations ont été introduites pour réduire l'utilisation de la mémoire lors de la restauration du dictionnaire de données logique lorsqu'il existe un grand nombre de tables.
- Correction d'un problème en raison duquel l'exécution de la `SHOW BINARY LOGS` commande prenait plus de temps sur un cluster où Enhanced Binlog était activé ou était activé précédemment. Ce problème pourrait également entraîner une augmentation de la latence de validation si plusieurs `SHOW BINARY LOGS` commandes étaient exécutées simultanément.
- Correction d'un problème de mémoire associé aux rôles par défaut du définisseur de vue.
- Correction d'un problème pouvant entraîner l'échec du processus de désactivation de la fonction de transfert d'écriture.
- Correction d'un problème qui empêchait l'établissement de nouvelles connexions client avec la base de données lorsque le transfert d'écriture est activé.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données Writer lorsqu'une instance de lecteur utilisant le transfert d'écriture exécute une instruction DML contenant une `timestamp` valeur et que le paramètre de `time_zone` base de données est défini sur « UTC ».
- Correction d'un problème qui provoquait l'indisponibilité intermittente d'un réplica Aurora Read ou des incohérences dans la définition de la table avec une erreur `'Table does not exist'` sur le réplica en raison de requêtes de lecture simultanées sur le réplica et d'opérations DDL sur le scripteur.
- Correction d'un problème qui pouvait entraîner un ensemble de résultats incomplet lors de l'exécution de requêtes impliquant `LEFT-` ou d'`RIGHT-JOIN`opérations utilisant l'algorithme de jointure par hachage avec Parallel Query.
- À partir de cette version d'Aurora MySQL, l'optimisation de l'insertion rapide n'est plus activée. Pour plus d'informations, consultez Améliorations des performances Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Mises à niveau et migrations

- Suppression des rôles par défaut créés inutilement lors de la mise à niveau de AMS2 vers AMS3.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.28 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème qui permet de résoudre le blocage lorsque `FLUSH STATUS` et `SHOW PROCESS LIST` sont exécutés simultanément. `COM_CHANGE_USER` (Insecte #35218030)

## Mises à jour du moteur de base de données Aurora MySQL 2024-06-26 (version 3.04.3, compatible avec MySQL 8.0.28)

Version : 3.04.3

Aurora MySQL 3.04.3 est généralement disponible. Les versions 3.04 d'Aurora MySQL sont compatibles avec MySQL 8.0.28. Pour plus d'informations sur les modifications apportées à la communauté, consultez les notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
> Nous vous recommandons de ne pas définir le `AutoMinorVersionUpgrade` paramètre sur `true` (ou de ne pas activer la mise à niveau automatique des versions mineures dans AWS Management Console) pour les versions LTS. Cela pourrait entraîner la mise à niveau de votre cluster de base de données vers la prochaine version cible pour la campagne de mise à niveau automatique des versions mineures, qui n'est peut-être pas une version LTS.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* et 3.07.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement disponible vers un cluster Aurora MySQL version 3.04.3.

Pour plus d'informations sur la planification d'une mise à niveau vers la version 3 d'Aurora MySQL, voir Planification d'une mise à niveau de version majeure pour un cluster de bases de données Aurora MySQL. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour obtenir des informations de dépannage, consultez la section Résolution des problèmes liés à la mise à niveau sur place d'Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs :

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 8.0.28. Les correctifs CVE suivants sont inclus :

- CVE-2024-0853

Améliorations de la disponibilité :

- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données Aurora MySQL lors de l'exécution d'une requête parallèle.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur lors de la lecture d'une table modifiée ou supprimée sur l'instance de base de données d'écriture.
- Correction d'un problème qui provoquait une violation de l'accès à la mémoire entraînant la libération d'un objet mutex n'appartenant plus au thread.

- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données Aurora MySQL Writer lorsqu'une session de transfert d'écriture était fermée lors de l'exécution d'une requête transférée.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données lors de la gestion de grands ensembles GTID sur une instance activée pour les journaux binaires.
- Correction d'un problème lors du traitement des `INSERT` requêtes sur les tables partitionnées InnoDB qui pouvait entraîner une diminution progressive de la mémoire libre dans l'instance de base de données.
- Correction d'un problème qui, dans de rares cas, pouvait provoquer le redémarrage d'une instance de lecteur lors de l'exécution de `SELECT` requêtes sur des tables soumises à une contrainte de clé étrangère.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une base de données lorsque la restauration du dictionnaire de données InnoDB prenait beaucoup de temps pendant la restauration de la base de données.
- Correction d'un problème qui provoquait le redémarrage d'une base de données lorsqu'une contrainte de clé `DELETE` étrangère `UPDATE` ou en cascade était définie sur une table où une colonne virtuelle était impliquée soit en tant que colonne dans la contrainte de clé étrangère, soit en tant que membre de la table référencée.
- Correction d'un problème Aurora Serverless v2 qui pouvait entraîner le redémarrage de la base de données lors de la mise à l'échelle.

Améliorations générales :

- Correction d'un problème qui fournissait une valeur incorrecte pour la variable d'`threads_running`état lors de l'utilisation de la base de données globale Aurora.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données en raison d'informations inexactes sur le détenteur du verrou `rw_lock` lors de l'utilisation de lectures parallèles.
- Correction d'un problème de gestion de la mémoire qui entraînait une diminution de la mémoire disponible au fil du temps lors de l'exécution de `SELECT ... INTO OUTFILE ...` requêtes.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lorsque le stockage local de l'instance de base de données atteignait sa pleine capacité.

- Correction d'un problème en raison duquel le schéma de performance n'était pas activé lorsque la gestion automatisée de Performance Insights était activée pour les instances de base de données db.t4g.medium et db.t4g.large.
- Correction d'un problème lié à l'application de correctifs sans interruption de service (ZDP) qui empêchait une instance de base de données de fermer les connexions client lorsqu'elle atteignait la configuration client de l'un ou de l'autre. `wait_timeout` `interactive_timeout`
- Problème résolu : une `SELECT` requête sur une instance de lecteur Aurora pouvait échouer en raison de l'inexistence de la table d'erreurs lorsque la table contient au moins un index de recherche en texte intégral (FTS) et qu'une `TRUNCATE` instruction est exécutée sur l'instance de base de données Aurora Writer.

Mises à niveau et migrations :

- Correction d'un problème qui entraînait l'échec des mises à niveau ou des migrations lorsque la version du moteur de base de données Aurora MySQL cible était la version 3.04.0 ou supérieure. Cela se produit lorsque le paramètre du `lower_case_table_names` cluster de base de données est défini sur et que le classement de base de données MySQL est incompatible avec les noms de table en minuscules. 1

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires, jusqu'à la version 8.0.28 incluse. Pour de plus amples informations, veuillez consulter Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

## Mises à jour du moteur de base de données Aurora MySQL 2024-03-15 (version 3.04.2, compatible avec MySQL 8.0.28)

Version : 3.04.2

Aurora MySQL 3.04.2 est généralement disponible. Les versions 3.04 d'Aurora MySQL sont compatibles avec MySQL 8.0.28. Pour plus d'informations sur les modifications apportées à la communauté, consultez les notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora

MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
> Nous vous recommandons de ne pas définir le `AutoMinorVersionUpgrade` paramètre sur `true` (ou de ne pas activer la mise à niveau automatique des versions mineures dans AWS Management Console) pour les versions LTS. Cela pourrait entraîner la mise à niveau de votre cluster de base de données vers la prochaine version cible pour la campagne de mise à niveau automatique des versions mineures, qui n'est peut-être pas une version LTS.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* et 3.06.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement disponible vers un cluster Aurora MySQL version 3.04.2.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs :

Les correctifs CVE suivants sont inclus dans cette version :

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

Améliorations de la disponibilité :

- Correction d'un problème en raison duquel une instance de base de données de réplication en lecture ne pouvait pas être lancée correctement lorsque la charge de travail de l'instance de base de données d'écriture était élevée.
- Correction d'un problème à cause duquel une instance de base de données Aurora MySQL Writer pouvait basculer en raison d'un défaut du composant qui communique avec le stockage Aurora. Le défaut est dû à une rupture de communication entre l'instance de base de données et le stockage sous-jacent suite à une mise à jour logicielle.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de l'exécution simultanée des instructions SHOW STATUS et PURGE BINARY LOGS. `PURGE BINARY LOG`Sest une instruction gérée qui est exécutée pour respecter la période de conservation du journal binaire configurée par l'utilisateur.
- Correction d'un problème qui, lors du redémarrage d'une instance de base de données, pouvait entraîner un redémarrage supplémentaire.
- Correction d'un problème de conflit de verrou causé par un thread de journalisation d'audit pouvant conduire à une utilisation élevée du processeur et à des dépassements de délai d'attente pour les applications clientes.
- Correction d'un problème en raison duquel une instance de base de données Aurora MySQL pouvait subir plusieurs redémarrages au démarrage de l'instance alors que de grands segments de restauration étaient initialisés.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de l'exécution d'une requête faisant référence à une fonction d'agrégation.

Améliorations générales :

- Correction d'un problème qui pouvait entraîner l'échec d'une requête parallel en raison de problèmes réseau transitoires lors de la lecture de données depuis le volume du cluster de base de données Aurora
- Correction d'un problème empêchant l'utilisateur d'interrompre une requête ou de définir des délais d'expiration de session pour les `performance_schema` requêtes.
- Problème résolu : la réplication du journal binaire (binlog) configurée pour utiliser des certificats SSL personnalisés (mysql.rds_import_binlog_ssl_material) pouvait échouer lorsque l'instance de réplication était en cours de remplacement d'hôte.
- Correction d'un problème lié à la gestion des fichiers journaux d'audit qui pouvait rendre les fichiers journaux inaccessibles pour le téléchargement ou la rotation et, dans certains cas, augmenter l'utilisation du processeur.
- Restauration des `AUTO_INCREMENT` clés optimisée pour réduire le temps nécessaire à la restauration des instantanés, à la point-in-time restauration et au clonage de clusters de bases de données contenant un grand nombre de tables dans la base de données.
- Correction d'un problème à cause duquel les instructions SQL faisant référence à certaines `performance_schema` tables pouvaient renvoyer une erreur en raison de l'absence de ces tables après la migration de Community MySQL vers les versions 3.04.0 et 3.04.1 d'Aurora MySQL.
- Correction d'un problème en raison duquel les petites instances de répliques en lecture pouvaient subir un délai de réplication accru après une mise à niveau à partir de versions d'Aurora MySQL inférieures à 2.11.*.
- Correction d'un problème qui pouvait provoquer des erreurs clés dupliquées pour les `AUTO_INCREMENT` colonnes utilisant des index décroissants après une restauration instantanée, un retour en arrière ou une opération de clonage de base de données.
- Correction d'un problème qui empêchait les modifications du paramètre de `table_open_cache` base de données de prendre effet tant que l'instance de base de données n'était pas redémarrée.
- Correction d'un problème qui empêchait l'instance de base de données du lecteur d'ouvrir une table, avec une erreur 1146. Ce problème se produit lors de l'exécution de certains types d'instructions DDL (Data Definition Language) en ligne alors que l'`INPLACE`algorithme est utilisé sur l'instance de base de données Writer.
- Correction d'un problème empêchant le redémarrage de l'instance pendant la mise à l'échelle d'Aurora Serverless v2 lorsque le processus de surveillance interne soumet par erreur des demandes de mise à l'échelle en double.

- Correction d'un problème qui pouvait provoquer le redémarrage de la base de données lorsque les consommateurs de journaux binaires (binlog) connectés utilisaient un serveur de réplication de journaux binaires dupliqué. IDs

Mises à niveau et migrations :

- Correction d'un problème qui pouvait entraîner l'échec des mises à niveau majeures vers Aurora MySQL version 3 en raison de la présence d'entrées orphelines pour des espaces disque logiques déjà supprimés dans les tables système InnoDB d'Aurora MySQL version 2.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 8.0.28 incluse, en plus des suivantes. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème en raison duquel la valeur de la ligne de cache pouvait être calculée de manière incorrecte, ce qui provoquait un échec lors du redémarrage de la base de données sur les instances basées sur Graviton. (Correctif de bogue communautaire #35479763)
- L'exécution répétée d'une routine stockée, ayant comme sous-requête une instruction SELECT contenant plusieurs ou `XOR` conditions `AND`/`OR`, a entraîné une consommation excessive et éventuellement un épuisement de la mémoire virtuelle. (Correctif de bogue communautaire #33852530)

## Mises à jour du moteur de base de données Aurora MySQL 13/11/2023 (version 3.04.1, compatible avec MySQL 8.0.28)

Version : 3.04.1

Aurora MySQL 3.04.1 est généralement disponible. Les versions 3.04 d'Aurora MySQL sont compatibles avec MySQL 8.0.28. Pour plus d'informations sur les modifications apportées à la communauté, consultez les notes de mise à jour de MySQL 8.0.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
> Nous vous recommandons de ne pas définir le `AutoMinorVersionUpgrade` paramètre sur `true` (ou de ne pas activer la mise à niveau automatique des versions mineures dans AWS Management Console) pour les versions LTS. Cela pourrait entraîner la mise à niveau de votre cluster de base de données vers la prochaine version cible pour la campagne de mise à niveau automatique des versions mineures, qui n'est peut-être pas une version LTS.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* et 3.05.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement disponible vers un cluster Aurora MySQL version 3.04.1.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

# Améliorations

Améliorations de la disponibilité :

- Correction d'un problème de redémarrage des instances de base de données Aurora MySQL utilisant une requête parallèle lorsqu'elles exécutent un grand nombre de requêtes parallèles simultanées.
- Correction d'un problème qui pouvait entraîner la restauration incorrecte de l'ensemble GTID exécuté sur un cluster de répliques de journaux binaires (binlog) lorsque le journal binaire amélioré était activé lorsqu'une source de journal binaire était définie sur ou. `gtid_mode` `ON` `ON_PERMISSIVE` Ce problème peut entraîner un redémarrage supplémentaire de l'instance d'enregistreur du cluster de réplica pendant la récupération, ou entraîner des résultats incorrects lors de l'interrogation du jeu de GTID exécuté.
- Correction d'un problème de gestion de la mémoire pouvant entraîner le redémarrage ou le basculement d'une instance de base de données Aurora MySQL en raison d'une diminution de la mémoire disponible lorsque le journal binaire amélioré est activé.
- Correction d'un problème susceptible d'entraîner le redémarrage de l'instance d'enregistreur lorsque celle-ci augmente le volume de la base de données jusqu'à un multiple de 160 Go.
- Correction d'un problème de blocage d'une instance de base de données MySQL Aurora avec la fonctionnalité de journal binaire amélioré activée pendant le démarrage de l'instance de base de données alors que le processus de récupération du journal binaire est en cours d'exécution.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données en raison d'un blocage pendant l'exécution simultanée des instructions `SHOW STATUS` et `PURGE BINARY LOGS`. L'instruction gérée purge binary logs est exécutée pour respecter la durée de conservation du journal binaire configurée par l'utilisateur.
- Correction d'un problème susceptible d'entraîner l'indisponibilité du cluster de base de données si l'instance d'enregistreur redémarre alors que la base de données crée ou supprime des déclencheurs sur des tables système internes.
- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données en raison d'un long délai d'attente du sémaphore lors de l'utilisation de la fonctionnalité de journal binaire amélioré sur un cluster doté d'un réplica Aurora.

Améliorations générales :

- Correction d'un problème susceptible d'entraîner l'indisponibilité de la base de données lorsque le journal binaire amélioré est activé sur un cluster de base de données Aurora Serverless v2 exécuté sur Aurora MySQL 3.04.0.
- Suppression des métadonnées de stockage inutilisées avant d'écrire dans Aurora Storage lorsque la fonctionnalité améliorée de journal binaire est activée. Cette suppression permet d'éviter certains scénarios de redémarrage ou de basculement de base de données en raison d'une latence d'écriture accrue due à l'augmentation du nombre d'octets transmis sur le réseau.
- Correction d'un problème d'absence de création de tables de schéma de performance spécifiques à Aurora lors d'une mise à niveau ou d'une migration.
- Correction d'un problème en raison duquel les `NumBinaryLogFiles` métriques affichées pouvaient CloudWatch afficher des résultats incorrects lorsque le journal binaire amélioré était activé.

Mises à niveau et migrations :

- Lors de la mise à niveau de MySQL 5.7 vers MySQL 8.0 avec un très grand nombre de tables dans une seule base de données, le serveur consommait trop de mémoire. Il a été constaté que, pendant le processus de vérification de la possibilité de mettre à niveau les tables, nous avons récupéré tous les `Table` objets du dictionnaire de données dès le départ, nous les avons traités et en avons extrait le nom, puis les avons ajoutés à `CHECK TABLE ... FOR UPGRADE`la liste. La recherche préalable de tous les objets n'était pas nécessaire dans ce cas, et contribuait grandement à la consommation de mémoire. Pour corriger ce problème, nous récupérons désormais un objet `Table` à la fois, nous effectuons toutes les vérifications nécessaires, nous récupérons son nom et nous libérons l'objet avant de passer au suivant. (Bogue n° 34526001)

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.28 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème pouvant entraîner une utilisation accrue du processeur en raison de la rotation des certificats TLS en arrière-plan (correctif du bogue n° 34284186)

# Mises à jour du moteur de base de données Aurora MySQL 31/07/2023 (version 3.04.0, compatible avec MySQL 8.0.28)

Version : 3.04.0

Aurora MySQL 3.04.0 est disponible. Les versions 3.04 d'Aurora MySQL sont compatibles avec MySQL 8.0.28, les versions 3.03 d'Aurora MySQL sont compatibles avec MySQL 8.0.26 et les versions 3.02 d'Aurora MySQL sont compatibles avec MySQL 8.0.23. Pour plus d'informations sur les changements apportés entre la version 8.0.23 et la version 8.0.28, consultez Notes de mise à jour de MySQL 8.0.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
> Nous vous recommandons de ne pas définir le `AutoMinorVersionUpgrade` paramètre sur `true` (ou de ne pas activer la mise à niveau automatique des versions mineures dans AWS Management Console) pour les versions LTS. Cela pourrait entraîner la mise à niveau de votre cluster de base de données vers la prochaine version cible pour la campagne de mise à niveau automatique des versions mineures, qui n'est peut-être pas une version LTS.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.11.1, 2.11.2, 3.01.*, 3.02.*, 3.03.* et 3.04.0.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement pris en charge vers un cluster Aurora MySQL version 3.04.0.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Le journal binaire amélioré Aurora MySQL n'est actuellement pas pris en charge par l'instance de base de données Aurora Serverless v2 sur Aurora MySQL version 3.04.0. L'activation de cette fonctionnalité peut entraîner l'indisponibilité de la base de données. Si vous avez besoin du journal binaire amélioré sur Aurora MySQL version 3.04.0, nous vous recommandons d'utiliser une classe d'instances de base de données non Serverless ou de définir les unités de capacité Aurora minimale et maximale l'instance de base de données Serverless v2 sur la même valeur.
> Pour plus d'informations sur le journal binaire amélioré dans Aurora MySQL, consultez le Guide de l'utilisateur d'Aurora.

## Améliorations

Nouvelles fonctions :

- Amélioration des performances des requêtes utilisant des index InnoDB Full-Text pour rechercher des phrases en mode langage naturel. Pour plus d'informations sur les recherches Full-Text dans MySQL, consultez Full-Text Search Functions.
- Amazon Aurora MySQL prend en charge le transfert d'écriture local (entre clusters) Vous pouvez désormais transférer les opérations d'écriture d'une instance de base de données de lecteur vers une instance de base de données d'enregistreur au sein d'un cluster de bases de données Aurora MySQL. Pour plus d'informations, consultez Utilisation du transfert d'écriture local dans un cluster de bases de données Amazon Aurora MySQL.

- Ajout de la possibilité de modifier la valeur du paramètre `aurora_replica_read_consistency` de la fonctionnalité Utilisation du transfert d'écriture dans une base de données Amazon Aurora globale dans les sessions dont le paramètre `autocommit` est désactivé. Pour plus d'informations, consultez Paramètres de configuration pour le transfert d'écriture.
- À partir d'Aurora MySQL 3.04, pour la fonctionnalité de transfert d'écriture dans une base de données globale, vous pouvez désormais définir la valeur du paramètre `aurora_replica_read_consistency` via le cluster de base de données et les groupes de paramètres d'instance de base de données. Avant Aurora MySQL version 3.04, la valeur de ce paramètre ne pouvait être configurée qu'au niveau de la session.

Problèmes de sécurité corrigés et CVEs :

- Le SSL/TLS fournisseur est passé d'AWS-LCOpenSSL à. Ce remplacement entraîne un certain nombre de changements, notamment mais sans s'y limiter :
  - Les connexions aux bases de données utilisant le protocole SSL peuvent désormais être restaurées par redémarrage et par application de correctifs sans temps d'arrêt lors de la mise à niveau d'Aurora MySQL 3.04.0 vers une version ultérieure.
  - Support de la version 3, qui TLSv1 inclut le support des chiffrements SSL TLS_AES_128_GCM_, TLS_AES_256_GCM_ et TLS_ 0_ 05_SHA256. SHA384 CHACHA2 POLY13 SHA256
  - Arrêt de la prise en charge des chiffrements DHE-RSA-* les moins sécurisés.

  Pour plus d'informations, consultez Utilisation de TLS avec les clusters de bases de données Aurora MySQL
- Ajout du privilège dynamique `SHOW_ROUTINE` qui permet à `rds_superuser_role` d'accéder aux définitions et aux propriétés de toutes les routines stockées, telles que les procédures et les fonctions stockées. Pour plus d'informations, consultez SHOW_ROUTINE.
- Correction d'un problème pouvant entraîner l'omission d'événements dans le journal d'audit lors de la rotation du fichier d'audit.
- Prise en charge du protocole TLS (Transport Layer Security) 1.3 sécurisé et performant, tout en préservant la compatibilité avec la version TLS 1.2.
- Les versions TLS TLSv1 et TLSv1 .1 sont devenues obsolètes dans la version 8.0.26 de la communauté MySQL et, par conséquent, dans Aurora MySQL 3.03. Ces protocoles ont été supprimés de la version 8.0.28 de Community MySQL 8.0.28 et, par conséquent, d'Aurora MySQL 3.04. Par défaut, tout client sécurisé qui ne peut pas communiquer via le protocole TLS 1.2

ou version ultérieure sera rejeté. Pour plus d'informations sur la connexion à vos instances de base de données à l'aide du protocole TLS, consultez Sécurité avec Amazon Aurora MySQL.

Les correctifs CVE suivants sont inclus dans cette version :

- 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

Améliorations de la disponibilité :

- Correction d'un problème qui pouvait entraîner le redémarrage de la base de données pendant une longue période de restauration de transactions.
- Correction d'un problème lié au chiffrement des événements des flux d'activité des bases de données, susceptible de provoquer le redémarrage de la base de données.
- Correction d'un problème de gestion de la mémoire causé par une mémoire insuffisante lors de l'initialisation du pool de tampons InnoDB au démarrage ou de la mise à l'échelle d'Aurora Serverless v2. Ce problème peut avoir entraîné des redémarrages d'instances de base de données ou une dégradation des performances, notamment une réduction du débit ou une augmentation de la latence.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de lecteur Aurora MySQL lors de l'exécution d'une requête utilisant un plan d'exécution parallèle de requêtes Aurora MySQL.
- Correction d'un problème qui, dans certaines situations, pouvait provoquer le redémarrage des instances du lecteur Aurora lors d'une estimation de plage.

- Correction d'un problème qui pouvait interrompre la restauration de la base de données au démarrage si le redémarrage se produisait lors de l'exécution d'opérations d'insertion intensives impliquant l'incrémentation automatique de colonnes.
- Correction d'un problème avec l'audit avancé Aurora qui entraîne une journalisation excessive des messages d'information dans le journal des erreurs d'Aurora MySQL lorsque la variable de serveur `server_audit_events` est définie sur `ALL` ou `QUERY`. Ce problème peut entraîner le redémarrage d'une instance de base de données.
- Correction d'un problème qui pouvait provoquer le redémarrage de la base de données lors de l'annulation d'une `INSERT` instruction lorsque la requête parallèle était activée.
- Correction d'un problème qui pouvait provoquer le redémarrage de l'instance de base de données lors de l'exécution de l'outil de `EXPLAIN ANALYZE` profilage sur une requête renvoyant le résultat `all select tables were optimized away` dans la colonne `EXTRA` d'informations. Pour plus d'informations, consultez Format de sortie `EXPLAIN` dans la documentation de MySQL.
- Correction d'un problème qui provoquait le redémarrage d'une instance de lecteur de région secondaire de la base de données globale Aurora utilisant le transfert d'écriture global lorsqu'une instruction de validation implicite transférée rencontrait une erreur.
- Correction d'un problème qui provoquait le redémarrage de l'instance d'écriture d'une région principale de base de données globale Aurora lorsqu'une `SELECT FOR UPDATE` requête était exécutée à l'aide du transfert d'écriture global depuis une région secondaire de base de données globale Aurora.

Améliorations générales :

- Ajout d'une nouvelle procédure stockée, `mysql.rds_gtid_purged`, pour permettre aux clients de définir la variable système `GTID_PURGED`. Pour plus d'informations, consultez mysql.rds_gtid_purged.
- Ajout de deux nouvelles procédures stockées, `mysql.rds_start_replication_until` et `mysql.rds_start_replication_until_gtid`, qui permettent aux clients de configurer une position pour l'arrêt de la réplication du journal binaire. Pour plus d'informations sur la configuration d'une position pour l'arrêt de la réplication du journal binaire dans Aurora MySQL, consultez mysql.rds_start_replication_until.
- Correction d'un problème empêchant les procédures stockées de contrôle de la réplication Aurora MySQL de modifier la variable `sql_log_bin` lorsqu'elle est appelée à partir d'une session dont le mode autocommit est désactivé.

- Ajout de la prise en charge de la réplication logique pour les instructions DCL (Data Control Language) suivantes : `GRANT/REVOKE` et `CREATE/DROP/ALTER/RENAME USER`.
- Correction d'un problème empêchant l'obsolescence des statistiques d'InnoDB, ce qui pouvait parfois générer un plan d'exécution de requêtes sous-optimal qui pouvait entraîner une augmentation de la durée d'exécution des requêtes.
- Ajout de deux nouvelles vues du système, `information_schema.aurora_global_db_instance_status` et `information_schema.aurora_global_db_status`. Ces vues peuvent être utilisées pour afficher le statut et la topologie des ressources principales et secondaires dans un cluster de base de données globale Aurora MySQL. Pour plus d'informations sur les deux vues du système, consultez Tables information_schema spécifiques à Aurora MySQL.
- Correction d'un problème empêchant un utilisateur d'accéder à la base de données en utilisant un caractère générique dans le nom de la base de données après avoir exécuté l'instruction `SET ROLE` avec un caractère générique d'échappement.
- Correction d'un problème susceptible d'empêcher l'écriture dans le journal d'audit d'événements signalés lors du traitement des rotations du journal d'audit.
- Correction d'un problème de création de table temporaire interne, via une exécution `TRIGGER`, qui peut entraîner le redémarrage d'une instance de base de données d'enregistreur.
- Ajout d'une nouvelle variable système, `innodb_aurora_max_partitions_for_range`. Si les statistiques persistantes ne sont pas disponibles, vous pouvez utiliser ce paramètre pour accélérer les estimations du nombre de lignes dans des tables partitionnées. Vous trouverez de plus amples informations dans la documentation intitulée Paramètres de configuration d'Aurora MySQL.
- Correction d'un problème autorisant à tort les clients à définir `ROW_FORMAT` sur `COMPRESSED` lors de la création de tables partitionnées. Les tables seront implicitement converties au format `COMPACT` et un avertissement indiquera qu'Aurora MySQL ne prend pas en charge les tables compressées.
- Correction d'un problème qui pouvait entraîner l'arrêt de la réplication des journaux binaires multithread lorsque la `replica_parallel_type` variable était définie sur `LOGICAL_CLOCK` et que la `replica_preserve_commit_order` variable était activée. `ON` Ce problème peut se produire lorsqu'une transaction supérieure à 500 Mo est exécutée sur la source.
- Correction d'un problème d'activation de la fonctionnalité de transfert d'écriture dans une base de données globale, qui peut entraîner le transfert involontaire des modifications apportées à la configuration `performance_schema` des instances de lecteur des régions secondaires vers l'instance d'enregistreur de la région principale.

- Correction d'un problème d'absence de mise à jour de la variable d'état du serveur `innodb_buffer_pool_reads` après la lecture d'une page de données dans le système de fichiers de stockage Aurora.
- Les requêtes parallèles d'Amazon Aurora MySQL ne sont pas prises en charge lors du choix de la configuration du cluster Aurora I/O-Optimized. Pour plus d'informations, consultez Limites des requêtes parallèles pour Amazon Aurora MySQL.
- Correction d'un problème d'activation des requêtes parallèles qui conduit l'optimiseur de plan de requêtes à choisir un plan d'exécution inefficace pour certaines requêtes `SELECT` bénéficiant d'un index principal ou secondaire.
- Mise à niveau des définitions de fuseaux horaires vers la version IANA 2023c.
- Optimisation des performances de la gestion des fichiers sur les réplicas de fichiers binaires afin de réduire les conflits d'écriture dans les fichiers journaux de relais.
- Correction d'un problème en raison duquel la `RPO_LAG_IN_MILLISECONDS` colonne du `information_schema.aurora_global_db_status` tableau et de la `AuroraGlobalDBRPOLag` CloudWatch métrique affichait toujours zéro, quelle que soit la charge de travail de l'utilisateur.
- Introduction d'un nouveau paramètre, `aurora_tmptable_enable_per_table_limit`. Lorsque ce paramètre est activé, la `tmp_table_size` variable définit la taille maximale de chaque table temporaire interne en mémoire créée par le moteur TempTable de stockage. Pour plus d'informations, consultez Moteur de stockage pour tables temporaires internes (implicites).
- Correction d'un problème d'établissement d'une connexion supplémentaire lorsque la fonctionnalité de transfert d'écriture dans une base de données globale est activée. Le problème se produit lorsque des transactions en lecture seule sur une instance de lecteur transmettent de manière incorrecte une instruction implicit commit à l'enregistreur.
- Correction d'un problème de renseignement des champs `PROCESSLIST_USER` et `PROCESSLIST_HOST` de la table `performance_schema.threads` sur l'enregistreur de la région principale pour les connexions utilisant la fonctionnalité de transfert d'écriture dans la base de données globale. Vous trouverez plus d'informations sur cette table et ce schéma de performance dans le manuel de référence MySQL, The threads Table, et le guide de l'utilisateur Présentation du schéma de performance.
- Correction d'un problème d'affichage de valeurs incorrectes dans la métrique `CommitLatency` CloudWatch pour les instances de lecteur des régions secondaires lorsque la fonctionnalité de transfert d'écriture dans la base de données globale est utilisée. Pour surveiller la latence des instructions DML transférées sur des clusters de bases de données secondaires, il est recommandé d'utiliser les métriques `ForwardingReplicaDMLLatency` et

`ForwardingWriterDMLLatency` La latence de la validation peut également être observée à l'aide de la métrique `CommitLatency` sur l'instance d'enregistreur de la région principale. De plus amples informations sont disponibles dans le guide de l'utilisateur d'Aurora, Amazon CloudWatch metrics for write forwarding.

- Correction d'un problème de signalement inapproprié d'erreurs par les procédures stockées de contrôle de la réplication Aurora MySQL utilisées pour gérer et configurer la réplication du journal binaire lorsque la réplication multithread dans le journal binaire est configurée avec une valeur supérieure à 0 pour la variable `replica_parallel_workers`.
- Correction d'un problème qui pouvait entraîner une consommation élevée du processeur lorsque plusieurs sessions tentaient d'accéder à une page qui n'existe pas en mémoire.

Mises à niveau et migrations :

- Pour effectuer la mise à niveau d'une version mineure d'une base de données globale Aurora MySQL version 3.01, 3.02 ou 3.03 vers Aurora MySQL version 3.04 ou ultérieure, consultez Mise à niveau d'Aurora MySQL par modification de la version du moteur.
- Correction d'un problème susceptible de faire échouer la prévérification de la mise à niveau en raison d'incohérences de schéma signalées pour les tables `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` et `mysql.slow_log` lors de la mise à niveau d'Aurora MySQL 2 vers Aurora MySQL 3. Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.
- Correction d'un problème pouvant provoquer l'échec de mises à niveau majeures lors de la mise à niveau vers Aurora MySQL version 3 si la définition d'un déclencheur contient un mot-clé réservé qui n'est pas entre guillemets.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.28 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème de déplacement d'un bloc de mémoire tampon contenant une page de table temporaire intrinsèque lors du parcours d'une page, provoquant un échec d'assertion (Bogue n° 33715694)
- InnoDB : empêcher les opérations DDL en ligne d'accéder à la out-of-bounds mémoire (bogue n° 34750489, bogue n° 108925)

- Correction d'un problème qui pouvait parfois produire des résultats de requête incorrects lors du traitement d'instructions SQL complexes composées de plusieurs expressions de table communes imbriquées (CTEs) (bogue n° 34572040, bogue n° 34634469, bogue n° 33856374)

# Mises à jour du moteur de base de données Aurora MySQL 08/2023 (version 3.03.3) (obsolète)

Version : 3.03.3

Aurora MySQL 3.03.3 est en disponibilité générale. Les versions 3.03 d'Aurora MySQL sont compatibles avec MySQL 8.0.26. Pour plus d'informations sur les changements apportés entre la version 8.0.23 et la version 8.0.28, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* et 3.05.*

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement disponible vers un cluster Aurora MySQL version 3.03.3.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

# Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs de CVE supplémentaires ci-dessous :

- CVE-2023-38545

Améliorations de la disponibilité :

- Correction d'un problème de redémarrage des instances de base de données Aurora MySQL utilisant une requête parallèle lorsqu'elles exécutent un grand nombre de requêtes parallèles simultanées.
- Correction d'un problème pouvant conduire à la récupération incorrecte du jeu de GTID exécuté sur un cluster de réplica du journal binaire avec le journal binaire amélioré activé lorsqu'une source binlog est définie sur `gtid_mode` `ON` ou `ON_PERMISSIVE`. Ce problème peut entraîner un redémarrage supplémentaire de l'instance d'enregistreur du cluster de réplica pendant la récupération, ou entraîner des résultats incorrects lors de l'interrogation du jeu de GTID exécuté.
- Correction d'un problème de gestion de mémoire qui pouvait entraîner le redémarrage ou le basculement d'une instance de base de données Aurora MySQL en raison d'une diminution de la quantité de mémoire pouvant être libérée lorsque le journal binaire amélioré était activé.
- Correction d'un problème susceptible d'entraîner le redémarrage de l'instance d'enregistreur lorsque celle-ci augmente le volume de la base de données jusqu'à un multiple de 160 Go.
- Correction d'un problème de blocage d'une instance de base de données MySQL Aurora avec la fonctionnalité de journal binaire amélioré activée pendant le démarrage de l'instance de base de données alors que le processus de récupération du journal binaire est en cours d'exécution.
- Correction d'un problème d'application de correctifs sans temps d'arrêt, qui provoquait le redémarrage d'une instance et la fermeture inattendue des connexions à la base de données.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données en raison d'un blocage pendant l'exécution simultanée des instructions `SHOW STATUS` et `PURGE BINARY LOGS`. L'instruction gérée purge binary logs est exécutée pour respecter la durée de conservation du journal binaire configurée par l'utilisateur.
- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données en raison d'un long délai d'attente du sémaphore lors de l'utilisation de la fonctionnalité de journal binaire amélioré sur un cluster doté d'un réplica Aurora.

Améliorations générales :

- Suppression des métadonnées de stockage inutilisées avant d'écrire sur le stockage Aurora lorsque la fonctionnalité de journal binaire amélioré est activée. Cette suppression permet d'éviter certains scénarios de redémarrage ou de basculement de base de données en raison d'une latence d'écriture accrue due à l'augmentation du nombre d'octets transmis sur le réseau.
- Correction d'un problème en raison duquel les `NumBinaryLogFiles` métriques affichées pouvaient CloudWatch afficher des résultats incorrects lorsque le journal binaire amélioré était activé.
- Correction d'un problème empêchant la modification du paramètre `table_open_cache` de la base de données de prendre effet tant que l'instance de base de données n'est pas redémarrée.
- Correction d'un problème qui pouvait provoquer le redémarrage de la base de données lorsque les consommateurs de journaux binaires (binlog) connectés utilisaient un serveur de réplication de journaux binaires dupliqué. IDs

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.26 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème pouvant entraîner une utilisation accrue du processeur en raison de la rotation des certificats TLS en arrière-plan (correctif du bogue n° 34284186)

## Mises à jour du moteur de base de données Aurora MySQL 29/08/2023 (version 3.03.2) (obsolète)

Version : 3.03.2

Aurora MySQL 3.03.2 est disponible. Les versions 3.04 d'Aurora MySQL sont compatibles avec MySQL 8.0.28, les versions 3.03 d'Aurora MySQL sont compatibles avec MySQL 8.0.26 et les versions 3.02 d'Aurora MySQL sont compatibles avec MySQL 8.0.23. Pour plus d'informations sur les changements apportés entre la version 8.0.23 et la version 8.0.28, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 3.01.*, 3.02.*, 3.03.* et 3.04.*.

Vous pouvez effectuer une mise à niveau sur place, restaurer un instantané ou lancer une mise blue/green à niveau gérée à l'aide d'Amazon RDS Blue/Green Deployments depuis n'importe quel cluster Aurora MySQL version 2 actuellement disponible vers un cluster Aurora MySQL version 3.03.2.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs :

- Correction d'un problème en raison duquel le journal d'audit pouvait manquer des événements lors de la rotation du fichier journal d'audit.

Les correctifs CVE suivants sont inclus dans cette version :

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

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

Améliorations de la disponibilité :

- Correction d'un problème qui pouvait entraîner le redémarrage de la base de données pendant une longue période de restauration de transactions.
- Correction d'un problème qui pouvait rendre le cluster de base de données indisponible lorsque l'instance du rédacteur redémarre alors que la base de données créait ou supprimait des déclencheurs sur les tables système internes.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de l'exécution d'une requête faisant référence à une fonction d'agrégation.
- Correction d'un problème qui pouvait provoquer le redémarrage de la base de données lors de l'annulation d'une `INSERT` instruction lorsque la requête parallèle était activée.
- L'insertion rapide n'est activée que pour les tables InnoDB classiques dans Aurora MySQL version 3.03.2 et supérieure. Cette optimisation ne fonctionne pas pour les tables temporaires InnoDB. Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème empêchant l'instance de lecteur d'ouvrir une table (`ERROR 1146`). Ce problème se produit lors de l'exécution de certains types d'instructions DDL (Data Definition Language) en ligne alors que l'algorithme `INPLACE` est utilisé sur l'instance d'enregistreur.
- Optimisation des performances de la gestion des fichiers sur les réplicas de fichiers binaires afin de réduire les conflits d'écriture dans les fichiers journaux de relais.
- Correction d'un problème lors duquel, lors de d'activation d'une requête parallèle, l'optimiseur de plan de requêtes choisissait un plan d'exécution inefficace pour certaines requêtes `SELECT` bénéficiant d'un index principal ou secondaire.
- Ajout de la prise en charge de la réplication logique pour les instructions DCL (Data Control Language) suivantes : `GRANT/REVOKE` et `CREATE/DROP/ALTER/RENAME USER`.
- Les requêtes parallèles d'Amazon Aurora MySQL ne sont pas prises en charge lors du choix de la configuration du cluster Aurora I/O-Optimized. Pour plus d'informations, consultez Limites des requêtes parallèles pour Aurora MySQL.

Mises à niveau et migrations :

- Pour effectuer la mise à niveau d'une version mineure d'une base de données globale Aurora MySQL version 3.01 ou 3.02 vers Aurora MySQL version 3.03 ou ultérieure, consultez Mise à niveau d'Aurora MySQL par modification de la version du moteur.
- Correction d'un problème pouvant provoquer l'échec de mises à niveau majeures lors de la mise à niveau vers Aurora MySQL version 3 si la définition d'un déclencheur contient un mot-clé réservé qui n'est pas entre guillemets.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.26 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème qui pouvait parfois produire des résultats de requête incorrects lors du traitement d'instructions SQL complexes composées de plusieurs expressions de table communes imbriquées (CTEs). (Bogues n° 34572040, 34634469 et 33856374)
- InnoDB : échec d'assertion provoqué par une condition de concurrence entre des threads tentant de désinitialiser et d'initialiser des statistiques pour la même table. (Bogue n° 33135425)
- InnoDB : Empêchez les opérations DDL en ligne d'accéder à la mémoire. out-of-bounds (Bogues n ° 34750489 et 108925)

## Mises à jour du moteur de base de données Aurora MySQL 11/05/2023 (version 3.03.1) (obsolète)

Version : 3.03.1

Aurora MySQL 3.03.1 est disponible. Les versions 3.03 d'Aurora MySQL sont compatibles avec MySQL 8.0.26, et les versions 3.02 d'Aurora MySQL sont compatibles avec MySQL 8.0.23. Pour plus d'informations sur les changements apportés entre la version 8.0.23 et la version 8.0.26, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, voir Comparaison entre Aurora MySQL version 2

et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.11.1, 2.11.2, 3.01.*, 3.02.* et 3.03.*.

Vous pouvez effectuer une mise à niveau sur place ou restaurer un instantané à partir d'un cluster Aurora MySQL version 2 actuellement pris en charge dans Aurora MySQL 3.03.1.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- La configuration I/O de stockage optimisée Aurora est disponible à partir de la version 3.03.1. Pour plus d'informations, consultez Configurations de stockage pour les clusters de bases de données Amazon Aurora.
- Ajout d'une nouvelle variable système, `innodb_aurora_max_partitions_for_range`. Si les statistiques persistantes ne sont pas disponibles, vous pouvez utiliser ce paramètre pour accélérer les estimations du nombre de lignes dans des tables partitionnées. Vous trouverez de plus amples informations dans la documentation intitulée Paramètres de configuration d'Aurora MySQL.

Améliorations de la disponibilité :

- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données en raison d'un accès incorrect à un emplacement de mémoire non valide lors de fermeture d'une connexion immédiatement après la validation d'une transaction.

- Correction d'un problème avec l'audit avancé Aurora qui entraîne une journalisation excessive des messages d'information dans le journal des erreurs d'Aurora MySQL lorsque la variable de serveur `server_audit_events` est définie sur ALL ou QUERY. Ce problème peut entraîner le redémarrage de l'instance de base de données.
- Correction d'un problème susceptible, dans certains cas, d'entraîner le redémarrage d'instances de lecteur Aurora lorsqu'elles tentaient de lire une page qui n'était plus accessible durant une estimation de plage.
- Correction d'un problème pouvant entraîner le redémarrage d'une instance de lecteur Aurora MySQL lors de l'exécution d'une requête utilisant un plan d'exécution de requêtes parallèles Aurora.
- Correction d'un problème lors duquel des instances de bases de données utilisant la réplication du journal binaire pouvaient constater une augmentation de l'utilisation du processeur et du nombre d'échecs de connexion lorsque plusieurs consommateurs de la réplication du journal binaire étaient connectés.
- Correction d'un problème de prise en compte de méthodes d'accès à l'analyse d'index non prises en charge pour les expressions de table courantes (CTE) lors de la matérialisation des tables temporaires intermédiaires, ce qui pouvait entraîner des comportements indésirables, notamment des redémarrages de la base de données ou des résultats de requête incorrects. Nous résolvons ce problème en évitant d'utiliser de telles méthodes d'accès non prises en charge par analyse d'index sur les tables à l'aide du moteur TempTable de stockage.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème susceptible de ralentir l'exécution de l'instruction SHOW BINARY LOGS, et par conséquent de réduire le débit de validation de la base de données.
- Correction d'un problème pouvant entraîner l'échec de l'exportation parallèle des tables utilisateur auxquelles des colonnes ont été ajoutées à l'aide de la fonctionnalité Instant ADD COLUMN.
- Correction d'un problème susceptible d'empêcher l'écriture dans le journal d'audit d'événements signalés lors du traitement des rotations du journal d'audit.
- Correction d'un problème pouvant entraîner l'épuisement de la mémoire disponible lors de l'exécution de requêtes sur la table INFORMATION_SCHEMA INNODB_TABLESPACES.

- Correction d'un problème autorisant à tort les clients à définir ROW_FORMAT sur COMPRESSED lors de la création de tables partitionnées. Les tables seront implicitement converties au format COMPACT et un avertissement indiquera qu'Aurora MySQL ne prend pas en charge les tables compressées.

Mises à niveau et migrations :

- Pour effectuer la mise à niveau d'une version mineure d'une base de données globale Aurora MySQL version 3.01 ou 3.02 vers Aurora MySQL version 3.03 ou ultérieure, consultez Mise à niveau d'Aurora MySQL par modification de la version du moteur.
- Correction d'un problème susceptible de faire échouer la prévérification de la mise à niveau en raison d'incohérences de schéma signalées pour les tables `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` et `mysql.slow_log` lors de la mise à niveau d'Aurora MySQL 2 vers Aurora MySQL 3. Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.26 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème de déplacement d'un bloc de mémoire tampon contenant une page de table temporaire intrinsèque lors du parcours d'une page, provoquant un échec d'assertion. (Bogue n° 33715694)

## Mises à jour du moteur de base de données Aurora MySQL : 01/01 (version 3.03.0) (obsolète)

Version : 3.03.0

Aurora MySQL 3.03.0 est disponible. Les versions 3.03 d'Aurora MySQL sont compatibles avec MySQL 8.0.26, et les versions 3.02 d'Aurora MySQL sont compatibles avec MySQL 8.0.23. Pour plus d'informations sur les changements apportés entre la version 8.0.23 et la version 8.0.26, consultez Notes de mise à jour de MySQL 8.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. Pour connaître les différences entre Aurora MySQL version 3 et Aurora MySQL version 2, consultez la comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3. Pour une comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition, voir Comparaison entre Aurora MySQL version 3 et MySQL 8.0 Community Edition.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.*, 3.01.*, 3.02.* et 3.03.*.

Vous pouvez effectuer une mise à niveau sur place ou restaurer un instantané à partir d'un cluster Aurora MySQL version 2 actuellement pris en charge dans Aurora MySQL 3.03.0.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

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

Améliorations de la disponibilité :

- Correction d'un problème lors duquel de grandes classes d'instances de base de données pouvaient rencontrer des problèmes lors du redémarrage en raison d'une initialisation du pool de mémoire tampon plus longue que prévu.
- Correction d'un problème de redémarrage d'une instance pendant son processus de récupération lorsque la journalisation binaire est activée.
- Correction d'un problème susceptible de provoquer des échecs de connexion sur des instances de lecteur lors de l'exécution d'instructions DCL (Data Control Language), par exemple `GRANT` et `REVOKE`, ou lors de l'établissement de nouvelles connexions sur l'instance d'enregistreur.
- Correction d'un problème d'utilisation inapproprié d'une requête parallèle pour des opérations DML (Data Manipulation Language), telles que des instructions `DELETE` et`UPDATE`, qui ne sont pas prises en charge actuellement, et qui entraînent le redémarrage de l'instance de base de données. Pour plus d'informations sur les opérations prises en charge dans les requêtes parallèles, consultez Limites des requêtes parallèles pour Amazon Aurora MySQL.
- Correction d'un problème susceptible, dans de rares cas, de provoquer le redémarrage d'un réplica Aurora lors de l'exécution simultanée d'opérations de mise à jour ou de charges de travail DDL (Data Definition Language) volumineuses sur l'instance d'enregistreur et d'opérations de lecture sur le même ensemble de tables sur le réplica Aurora.
- Correction d'un problème avec l'opération de réduction d'échelle de l'instance de lecteur Aurora Serverless v2 qui pouvait entraîner le redémarrage de cette dernière et, dans de rares cas, une incohérence des données.
- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données en raison d'un accès incorrect à un emplacement de mémoire non valide lors de la fermeture d'une connexion à l'instance de la base de données.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de la base de données lors du traitement d'une requête avec une clause `GROUP BY` qui tronque une colonne décimale à zéro décimale.
- Correction d'un problème pouvant provoquer le redémarrage d'une instance de base de données en raison d'un accès incorrect à un enregistrement lors de l'exécution d'une requête de plage utilisant un index spatial.
- Correction d'un problème pouvant provoquer le redémarrage d'une instance de base de données sur des instances de réplica Aurora MySQL lorsque les tables temporaires internes dépassent les valeurs memory ou mmap par défaut ou configurées par le client.
- Correction d'un problème de gestion de mémoire causé par la rotation du journal d'audit avancé.

- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Amélioration de la latence des requêtes de lecture des sessions de transfert d'écriture de la base de données globale utilisant le paramètre de cohérence de lecture GLOBAL.
- Correction d'un problème lors duquel la valeur du paramètre `wait_timeout` n'était pas respectée après l'exécution des commandes `reset_connection` ou `change_user` par une session client.
- Correction d'un problème d'augmentation de la latence des applications lors de la connexion à une instance de base de données si celle-ci subissait une augmentation soudaine du nombre de connexions entrantes. Deux nouvelles CloudWatch métriques, `AuroraSlowHandshakeCount` et`AuroraSlowConnectionHandleCount`, ont été introduites pour aider à résoudre les retards d'établissement de connexion pour les instances de base de données Aurora MySQL. Vous trouverez de plus amples informations sur ces métriques dans la documentation relative aux définitions CloudWatch des métriques Aurora, relative aux CloudWatch métriques Amazon pour Amazon Aurora.
- Le paramètre `temptable_use_mmap` est devenu obsolète et ne devrait plus être pris en charge dans une future version de MySQL. Pour plus d'informations, consultez Moteur de stockage pour tables temporaires internes (implicites).
- Correction d'un problème susceptible d'entraîner des temps d'exécution plus longs que prévu pour l'instruction `SHOW BINARY LOGS`, et par conséquent une baisse du débit de validation de la base de données.

Mises à niveau et migrations :

- Pour effectuer la mise à niveau d'une version mineure d'une base de données globale Aurora MySQL version 3.01 ou 3.02 vers Aurora MySQL version 3.03 ou ultérieure, consultez Mise à niveau d'Aurora MySQL par modification de la version du moteur.
- Correction d'un problème susceptible de faire échouer les mises à niveau majeures d'Aurora MySQL version 2 vers Aurora MySQL version 3 lorsque le cluster contient un grand nombre de tables (plus de 750 000).

- Correction d'un problème susceptible de échouer les mises à niveau majeures d'Aurora MySQL version 2 vers Aurora MySQL version 3, car la migration des tables `mysql.innodb_table_stats` et `mysql.innodb_index_stats` prenait plus de temps que prévu. Ce problème concernait principalement les clusters de bases de données contenant des millions de tables.
- Correction d'un problème susceptible de faire échouer la mise à niveau d'Aurora MySQL 2 vers Aurora MySQL 3 en raison d'incohérences de schéma. Ces erreurs sont signalées par le pré-vérificateur de mise à niveau pour les tables `mysql.general_log_template` et `mysql.slow_log_template`. Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.
- Correction d'un problème susceptible de faire échouer les mises à niveau d'Aurora MySQL version 2 vers Aurora MySQL version 3 en raison de l'erreur`schemaInconsistencyCheck`. Cette erreur est due à des incohérences de schéma dans la table `mysql.table_migration_index_info`, comme indiqué par `upgrade-prechecks.log`. Pour plus d'informations sur la résolution des problèmes de mise à niveau vers Aurora MySQL version 3, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.26 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème lors duquel certains types de colonnes, dont `JSON` et `TEXT`, épuisaient parfois la mémoire tampon de tri si sa taille n'était pas au moins 15 fois supérieure à celle de la plus grande ligne du tri. Désormais, le tampon de tri ne doit être que 15 fois supérieur à la plus grande clé de tri. (Bogues n° 103325, 105532, 32738705 et 33501541)
- Correction d'un problème lors duquel InnoDB ne gérait pas toujours correctement certains noms légaux de partitions de table. (Bogue n° 32208630)
- Correction d'un problème susceptible, dans certaines conditions, de renvoyer des résultats incorrects en raison d'un calcul inexact de la propriété de nullabilité lors de l'exécution d'une requête avec une condition `OR`. (Bogue n° 34060289)
- Correction d'un problème susceptible, dans certaines conditions, de renvoyer des résultats incorrects lorsque les deux conditions suivantes sont remplies :

- une table dérivée est fusionnée dans le bloc de requête externe
- la requête inclut une jointure gauche et une `IN` sous-requête

(Bogue n° 34060289)

- Correction d'un problème de génération de valeurs `AUTO_INCREMENT` incorrectes lorsque la valeur maximale de la colonne de type entier était dépassée. L'erreur était due au fait que la valeur maximale de la colonne n'était pas prise en compte. La précédente valeur `AUTO_INCREMENT` valide aurait dû être renvoyée dans ce cas, ce qui aurait provoqué une erreur de doublon de clé. (Bogues n° 87926 et 26906787)
- Correction d'un problème de révocation du privilège `DROP` sur le schéma de performance. (Bogue n° 33578113)
- Correction d'un problème lors duquel une procédure stockée contenant une instruction IF utilisant EXISTS, qui agissait sur une ou plusieurs tables supprimées et recréées entre les exécutions, ne s'exécutait pas correctement après la première invocation. (Bogue n° 32855634)
- Correction d'un problème lors duquel une requête référence une vue dans une sous-requête et un bloc de requête externe peut provoquer un redémarrage inattendu. (Bogue n° 32324234)

# Mises à jour du moteur de base de données Aurora MySQL 17/04/2023 (version 3.02.3) (obsolète)

Version : 3.02.3

Aurora MySQL 2.02.3 est disponible. Les versions 3.02 d'Aurora MySQL sont compatibles avec MySQL 8.0.23, et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

Pour en savoir plus sur les nouvelles fonctionnalités de la version 3 d'Aurora MySQL et sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2 ou MySQL 8.0 communautaire, consultez la section Comparaison entre Aurora MySQL version 2 et Aurora MySQL version 3 dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.1, 2.11.2, 3.01.*, 3.02.* et 3.03.*.

Vous pouvez effectuer une mise à niveau sur place ou restaurer un instantané à partir d'un cluster Aurora MySQL version 2 actuellement pris en charge dans Aurora MySQL 3.02.3

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour la procédure de mise à niveau en elle-même, consultez Mise à niveau vers Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Améliorations de la disponibilité :

- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données en raison d'un accès incorrect à un emplacement de mémoire non valide lors de fermeture d'une connexion immédiatement après la validation d'une transaction.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème de prise en compte de méthodes d'accès à l'analyse d'index non prises en charge pour les expressions de table courantes (CTE) lors de la matérialisation des tables temporaires intermédiaires, ce qui pouvait entraîner des comportements indésirables, notamment des redémarrages de la base de données ou des résultats de requête incorrects. Ce problème a été résolu en évitant d'utiliser de telles méthodes d'accès à l'analyse d'index non prises en charge sur les tables à l'aide du moteur TempTable de stockage.
- Correction d'un problème susceptible, dans certaines conditions, de provoquer le redémarrage d'une instance de lecteur Aurora MySQL lors de l'accès à une table comportant une mise à jour importante ou des opérations DDL (Data Definition Language) exécutées simultanément sur l'instance d'enregistreur Aurora MySQL.

- Correction d'un problème susceptible, dans certaines conditions, d'entraîner le redémarrage d'instances de lecteur Aurora MySQL lors d'une tentative de lecture d'une page qui n'est plus accessible durant une estimation de plage.
- Correction d'un problème lors duquel des instances de bases de données utilisant la réplication du journal binaire pouvaient constater une augmentation de l'utilisation du processeur et du nombre d'échecs de connexion lorsque plusieurs consommateurs de la réplication du journal binaire étaient connectés.
- Correction d'un problème pouvant entraîner le redémarrage d'une instance de lecteur Aurora MySQL lors de l'exécution d'une requête utilisant un plan d'exécution de requêtes parallèles Aurora.

# Mises à jour du moteur de base de données Aurora MySQL le 18 novembre 2018 (version 3.02.2) (obsolète)

Version : 3.02.2

Aurora MySQL 2.02.2 est disponible. Les versions 3.02 d'Aurora MySQL sont compatibles avec MySQL 8.0.23, les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Pour en savoir plus sur les nouvelles fonctionnalités d'Aurora MySQL version 3 et sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2 ou MySQL 8.0 communautaire, consultez Comparaison d'Aurora MySQL version 2 et Aurora MySQL version 3 dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'un cluster Aurora MySQL version 2 actuellement pris en charge dans Aurora MySQL 3.02.2.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour la procédure de mise à niveau en elle-même, consultez Mise à niveau vers Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Aurora MySQL version 3.02.2 est disponible et compatible avec Community MySQL 8.0.23.

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

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

- CVE-2022-21444

Améliorations de la disponibilité :

- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données en raison d'un accès incorrect à un emplacement de mémoire non valide lors de la fermeture explicite ou implicite d'une connexion à l'instance de base de données.
- Correction d'un problème pouvant interrompre de façon répétée le démarrage de la base de données sur de grandes classes d'instances en raison d'une initialisation du pool de mémoire tampon plus longue que prévu.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage d'une instance de base de données lorsqu'Aurora Serverless v2 tente de manière incorrecte de mettre à jour le cache de la table lors de la mise à l'échelle.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de la base de données lors du traitement d'une requête avec une clause GROUP BY qui tronque une colonne décimale à zéro décimale.

- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème susceptible de faire échouer la mise à niveau d'Aurora MySQL version 2 (compatible avec MySQL 5.7) vers Aurora MySQL version 3 (compatible avec MySQL 8.0) en raison d'une incohérence des métadonnées dans la table mysql.host.
- Amélioration des performances afin de réduire la durée de la mise à niveau d'Aurora MySQL version 2 (compatible avec MySQL 5.7) vers Aurora MySQL version 3 (compatible avec MySQL 8.0). En parallélisant certaines étapes de la mise à niveau, la durée est encore plus courte lorsque de grandes classes d'instance, telles que db.r6g.16xlarge ou db.r5.24xlarge, sont utilisées.
- Ajout de la prise en charge de l'affichage de toutes les erreurs lors de la mise à niveau d'Aurora MySQL (compatible avec MySQL 5.7) vers Aurora MySQL version 3 (compatible avec MySQL 8.0) alors l'affichage était limité à 50 erreurs dans les versions précédentes.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner l'inexactitude des compteurs d'incrémentation automatique après une mise à niveau de la version majeure d'Aurora MySQL version 2 (compatible avec MySQL 5.7) vers Aurora MySQL version 3 (compatible avec MySQL 8.0).
- Correction d'un problème susceptible de faire échouer les mises à niveau majeures d'Aurora MySQL version 2 vers Aurora MySQL version 3, car la migration des tables `mysql.innodb_table_stats` et `mysql.innodb_index_stats` prenait plus de temps que prévu. Ce problème concernait principalement les clusters de bases de données contenant un grand nombre de tables (>1,5 million).
- Correction d'un problème susceptible de faire échouer les mises à niveau des versions majeures d'Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'un défaut dans le flux de mise à niveau du moteur AMS 8.0, qui entraînait l'accumulation des enregistrements de journaux sur le volume du cluster de stockage Aurora et interrompait les opérations d'écriture normales. Ce problème concernait principalement les clusters de bases de données contenant un grand nombre de tables (environ >750 000).
- Correction d'un problème qui empêchait les instances inactives d'Aurora MySQL Serverless v2 de passer à 0,5 ACUs car les threads de purge MySQL étaient incorrectement maintenus actifs.

- Correction d'un problème de latence accrue des applications lors de la connexion à une instance de base de données lorsque celle-ci subissait une augmentation soudaine du nombre de connexions entrantes.
- Introduction de deux nouvelles CloudWatch métriques Amazon pour aider à résoudre les retards d'établissement de connexion pour les instances de base de données Aurora MySQL. Vous trouverez plus d'informations AuroraSlowHandshakeCount et de AuroraSlowConnectionHandleCount statistiques dans les définitions CloudWatch des métriques Aurora.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.23 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'un problème susceptible, dans certaines conditions, de renvoyer des résultats incorrects en raison d'un calcul inexact de la propriété de nullabilité lors de l'exécution d'une requête avec une condition OR. (Bogue n° 34060289)
- Correction d'un problème susceptible, dans certaines conditions, de renvoyer des résultats incorrects lorsque les deux conditions suivantes sont remplies :
  - Une table dérivée est fusionnée dans le bloc de requête externe.
  - La requête inclut une jointure gauche et une sous-requête IN. (Bogue n° 34060289)
- Correction d'un problème empêchant la révocation du privilège DROP sur le schéma de performance. (Bogue n° 33578113)
- Correction d'un problème lors duquel une procédure stockée contenant une instruction IF utilisant EXISTS, qui agissait sur une ou plusieurs tables supprimées et recréées entre les exécutions, ne s'exécutait pas correctement lors des invocations suivantes après la première. (Bogue MySQL n ° 32855634).
- Correction d'un problème de génération de valeurs AUTO_INCREMENT incorrectes lorsque la valeur maximale de la colonne de type entier était dépassée. L'erreur était due à l'absence de prise en compte de la valeur maximale de colonne. La précédente valeur AUTO_INCREMENT valide aurait dû être renvoyée dans ce cas, ce qui a provoqué un doublon de clé. (Bogues n° 87926 et 26906787)
- Correction d'un problème qui pouvait entraîner un échec lors de la mise à niveau d'un cluster de base de données Aurora MySQL version 1 (compatible avec MySQL 5.6) contenant une table

créée par l'utilisateur avec une certaine table IDs. L'attribution de ces tables IDs peut entraîner un conflit entre les tables du dictionnaire de données IDs lors de la mise à niveau d'Aurora MySQL version 2 (compatible avec MySQL 5.7) vers Aurora MySQL version 3 (compatible avec MySQL 8.0) (Bug #33919635)

# Mises à jour du moteur de base de données Aurora MySQL 07.09.07 (version 3.02.1) (obsolète)

Version : 3.02.1

Aurora MySQL 3.02.1 est disponible. Les versions 3.02 d'Aurora MySQL sont compatibles avec MySQL 8.0.23, les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Pour en savoir plus sur les nouvelles fonctionnalités d'Aurora MySQL version 3 et sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2 ou MySQL 8.0 communautaire, consultez la section Comparaison d'Aurora MySQL version 2 et Aurora MySQL version 3 dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour la procédure de mise à niveau en elle-même, consultez Mise à niveau vers Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Aurora MySQL version 3.02.1 est disponible et compatible avec Community MySQL 8.0.23.

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- CVE-2022-0778

Améliorations de la disponibilité :

- Correction d'un problème pouvant provoquer un échec de connexion et une latence élevée lorsque plusieurs réplicas du journal binaire MySQL sont attachés à un nœud d'écriture Aurora ou qu'un grand nombre de requêtes de longue durée simultanées sont associées à un surcroît de nouvelles demandes de connexion.
- Correction d'un problème provoquant un redémarrage de base de données lorsque l'audit avancé des événements `CONNECT` est activé.
- Correction d'un problème provocant le redémarrage d'une base de données sur des instances de réplica en lecture Aurora MySQL lorsque les tables temporaires internes épuisent la mémoire allouée et que les fichiers mmap sont définis comme une valeur par défaut ou configurée par le client.
- Correction d'un problème pouvant provoquer le redémarrage répété d'un réplica en lecture pendant des opérations DDL simultanées sur des procédures stockées.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Ajout de la prise en charge des instances R6i.

Informations supplémentaires :

- Aurora MySQL version 3.02.1 ne prend pas en charge les mises à niveau des versions majeures directement à partir de la version 2 d'Aurora MySQL (compatible avec MySQL 5.7). Pour effectuer la mise à niveau d'une version majeure vers cette version, effectuez d'abord la mise à niveau de la

version majeure vers Aurora MySQL version 3.02.0, puis effectuez une mise à niveau sur place de la version mineure vers Aurora MySQL version 3.02.1.

# Mises à jour du moteur de base de données Aurora MySQL 20/04/2020 (version 3.02.0) (obsolète)

Version : 3.02.0

Aurora MySQL 3.02.0 est disponible. Les versions 3.02 d'Aurora MySQL sont compatibles avec MySQL 8.0.23, les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Pour en savoir plus sur les nouvelles fonctionnalités d'Aurora MySQL version 3 et sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2 ou MySQL 8.0 communautaire, consultez la section Comparaison d'Aurora MySQL version 2 et Aurora MySQL version 3 dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'un cluster Aurora MySQL version 2 actuellement pris en charge dans Aurora MySQL 3.02.0.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour la procédure de mise à niveau en elle-même, consultez Mise à niveau vers Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

# Améliorations

Aurora MySQL version 3.021.0 est disponible et compatible avec Community MySQL 8.0.23.

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- CVE-2021-22946

Nouvelles fonctions :

- Amazon Aurora Serverless v2 est disponible. Pour plus d'informations, consultez la présentation d'Amazon Aurora Serverless, le blog et la documentation Utiliser Aurora Serverless v2. Commencez dès aujourd'hui en créant une base de données Aurora Serverless v2 en quelques étapes seulement dans la AWS Management Console.

Améliorations de la disponibilité :

- Correction d'un problème pouvant entraîner un redémarrage en boucle et une indisponibilité du serveur lors de la suppression d'un enregistrement ou d'une table contenant au moins deux colonnes de longueur variable (types VARCHAR, VARBINARY, BLOB et TEXT). Pour plus de détails sur les types de colonnes, consultez innodb-row-format.
- Correction d'un problème lors duquel les connexions existantes expiraient et de nouvelles connexions ne pouvaient pas être établies sur un cluster avec le journal binaire activé et auquel au moins un consommateur du journal binaire était connecté, ce qui entraînait un conflit de ressources entre l'application et le ou les consommateurs.
- La mémoire disponible est indiquée par la `FreeableMemory` CloudWatch métrique. Pour plus d'informations, consultez CloudWatch les métriques Amazon pour Amazon Aurora.
  - Correction d'un problème pouvant entraîner le redémarrage ou le basculement d'une instance de base de données en raison d'une diminution de la mémoire disponible lorsque la réplication du journal binaire est activée.
  - Correction d'un problème pouvant provoquer le redémarrage ou le basculement d'une instance de base de données en raison d'une diminution de la mémoire disponible lors de la définition des variables de session.

  - Correction d'un problème pouvant provoquer le redémarrage ou le basculement d'une instance de base de données en raison d'une diminution de la mémoire disponible lorsque le processus de base de données ouvre un fichier existant.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner une erreur de double saisie lors de l'insertion de nouvelles lignes dans une table contenant une colonne `AUTO_INCREMENT` sur un cluster restauré à partir d'un instantané.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème lors duquel l'état du volume ne s'affichait pas lors de l'utilisation de la commande `SHOW VOLUME STATUS`. Pour plus d'informations, consultez AuroraMySQL.Managing.VolumeStatus.
- Correction d'un problème qui pouvait entraîner l'échec des appels à mysql_rds_import_binlog_ssl_material avec l'erreur de serveur MySQL 3512.
- Correction d'un problème de mauvais signalement du décalage des réplicas Aurora pour les instances de lecteur Aurora supprimées.

Mises à niveau/migration :

- Correction d'un problème pouvant faire échouer la migration de bases de données MySQL 8.0.x vers Aurora MySQL version 3 en raison d'un problème de copie des fichiers ibdata et des tablespaces dans le stockage Aurora.
- Correction d'un problème pouvant faire échouer la mise à niveau des clusters d'Aurora MySQL version 2 vers Aurora MySQL version 3 lorsque les tables de la base de données contenaient une grande quantité de données.
- Correction d'un problème pouvant faire échouer la restauration de clusters d'Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'un échec lors de la création des informations de dictionnaire de données sérialisées (SDI) pour une table.
- Correction d'un problème susceptible de faire échouer la mise à niveau d'Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'incohérences de schéma signalées par les prévérifications de mise à niveau des tables système RDS.

- Correction d'un problème pouvant faire échouer la migration ou la restauration de bases de données RDS for MySQL 8.0 ou Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'une syntaxe non valide dans une procédure stockée gérée par RDS.
- Correction d'un problème susceptible de faire échouer la mise à niveau d'Aurora MySQL 2 vers Aurora MySQL 3 en raison d'incohérences de schéma signalées par les prévérifications de mise à niveau pour les tables general log et slow log.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.23 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'une mauvaise gestion des tables temporaires utilisées pour les curseurs dans des procédures stockées qui pouvait entraîner un comportement inattendu du serveur, mysqld-8-0-24-bug. (Bogue n° 32416811)

## Mises à jour du moteur de base de données Aurora MySQL 15/04/15 (version 3.01.1) (obsolète)

Version : 3.01.1

Aurora MySQL 3.01.1 est disponible. Les versions 3.01 d'Aurora MySQL sont compatibles avec MySQL 8.0.23, les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Pour en savoir plus sur les nouvelles fonctionnalités d'Aurora MySQL version 3 et sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2 ou MySQL 8.0 communautaire, consultez Comparaison d'Aurora MySQL version 2 et Aurora MySQL version 3 dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'un cluster Aurora MySQL version 2 actuellement pris en charge dans Aurora MySQL 3.01.1.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour la procédure de mise à niveau en elle-même, consultez Mise à niveau vers Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Aurora MySQL version 3.01.1 est généralement disponible et compatible avec Community MySQL 8.0.23.

Aurora MySQL version 3.01.1 est recommandée pour les mises à niveau et les migrations vers une base de données Aurora compatible avec MySQL 8.0.

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

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

Améliorations de la disponibilité :

- La mémoire disponible est indiquée par la `FreeableMemory` CloudWatch métrique. Pour plus d'informations, consultez CloudWatch les métriques Amazon pour Amazon Aurora.
  - Correction d'un problème pouvant entraîner le redémarrage ou le basculement d'une instance de base de données en raison d'une diminution de la mémoire disponible lorsque la réplication du journal binaire est activée.

  - Correction d'un problème pouvant provoquer le redémarrage ou le basculement d'une instance de base de données en raison d'une diminution de la mémoire disponible lors de la définition des variables de session.
  - Correction d'un problème pouvant provoquer le redémarrage ou le basculement d'une instance de base de données en raison d'une diminution de la mémoire disponible lorsque le processus de base de données ouvre un fichier existant.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner une erreur de double saisie lors de l'insertion de nouvelles lignes dans une table contenant une colonne `AUTO_INCREMENT` sur un cluster restauré à partir d'un instantané.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème lors duquel l'état du volume ne s'affichait pas lors de l'utilisation de la commande `SHOW VOLUME STATUS`. Pour plus d'informations, consultez AuroraMySQL.Managing.VolumeStatus.
- Correction d'un problème qui pouvait entraîner l'échec des appels à mysql_rds_import_binlog_ssl_material avec l'erreur de serveur MySQL 3512.
- Correction d'un problème de mauvais signalement du décalage des réplicas Aurora pour les instances de lecteur Aurora supprimées.

Mises à niveau/migration :

- Correction d'un problème pouvant faire échouer la migration de bases de données MySQL 8.0.x vers Aurora MySQL version 3 en raison d'un problème de copie des fichiers ibdata et des tablespaces dans le stockage Aurora.
- Correction d'un problème pouvant faire échouer la mise à niveau des clusters d'Aurora MySQL version 2 vers Aurora MySQL version 3 lorsque les tables de la base de données contenaient une grande quantité de données.
- Correction d'un problème pouvant faire échouer la restauration de clusters d'Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'un échec lors de la création des informations de dictionnaire de données sérialisées (SDI) pour une table.

- Correction d'un problème susceptible de faire échouer la mise à niveau d'Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'incohérences de schéma signalées par les prévérifications de mise à niveau des tables système RDS.
- Correction d'un problème pouvant faire échouer la migration ou la restauration de bases de données RDS for MySQL 8.0 ou Aurora MySQL version 2 vers Aurora MySQL version 3 en raison d'une syntaxe non valide dans une procédure stockée gérée par RDS.
- Correction d'un problème susceptible de faire échouer la mise à niveau d'Aurora MySQL 2 vers Aurora MySQL 3 en raison d'incohérences de schéma signalées par les prévérifications de mise à niveau pour les tables general log et slow log.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 8.0.23 incluse. Pour plus d'informations, consultez Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x.

- Correction d'une mauvaise gestion des tables temporaires utilisées pour les curseurs dans des procédures stockées qui pouvait entraîner un comportement inattendu du serveur, mysqld-8-0-24-bug. (Bogue n° 32416811)

## Mises à jour du moteur de base de données Aurora MySQL 2021-11-18 (version 3.01.0) (obsolète)

Version : 3.01.0

Aurora MySQL 3.01.0 est disponible. Les versions 3.01 d'Aurora MySQL sont compatibles avec MySQL 8.0.23, les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Pour en savoir plus sur les nouvelles fonctionnalités d'Aurora MySQL version 3 et sur les différences entre Aurora MySQL version 3 et Aurora MySQL version 2 ou MySQL 8.0 communautaire, consultez Comparaison d'Aurora MySQL version 2 et Aurora MySQL version 3 dans le guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'un cluster Aurora MySQL version 2 actuellement pris en charge dans Aurora MySQL 3.01.0.

Pour en savoir plus sur la planification d'une mise à niveau vers Aurora MySQL version 3, consultez Planification de la mise à niveau d'Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour la procédure de mise à niveau en elle-même, consultez Mise à niveau vers Aurora MySQL version 3 dans le Guide de l'utilisateur Amazon Aurora. Pour obtenir des informations générales sur les mises à niveau d'Aurora MySQL, consultez Mise à niveau des clusters de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la résolution des problèmes, consultez Résolution des problèmes de mise à niveau avec Aurora MySQL version 3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Aurora MySQL version 3.01.0 est généralement compatible avec Community MySQL 8.0.23. Cette version inclut les correctifs de sécurité relatifs aux vulnérabilités et expositions (CVE) à partir de Community MySQL 8.0.23.

Aurora MySQL version 3.01.0 contient tous les correctifs de bogue spécifiques à Aurora via Aurora MySQL version 2.10.0.

Pour plus d'informations sur les nouvelles fonctionnalités d'Aurora MySQL version 3, consultez Fonctions de MySQL 8.0 Community Edition et Nouvelles optimisations des requêtes parallèles dans le Guide de l'utilisateur Amazon Aurora.

Améliorations de la disponibilité :

- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

# Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 2.

> **Important**
>
> La version 2 d'Aurora MySQL a atteint la fin du support standard le 31 octobre 2024. Pour plus d'informations, consultez Préparation à la fin du support standard d'Amazon Aurora MySQL Compatible Edition version 2.

Les mises à jour du moteur de base de données Amazon Aurora version 2 sont les suivantes.

- Mises à jour du moteur de base de données Aurora MySQL 2025-04-09 (version 2.12.5, compatible avec MySQL 5.7.44) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 2024-10-23 (version 2.12.4, compatible avec MySQL 5.7.44) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 2024-07-09 (version 2.12.3, compatible avec MySQL 5.7.44) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 2024-03-19 (version 2.12.2, compatible avec MySQL 5.7.44) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 28/12/2023 (version 2.12.1, compatible avec MySQL 5.7.40) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 25/10/2023 (version 2.12.0.1, compatible avec MySQL 5.7.40) - Version RDS Extended Support (bêta)
- Mises à jour du moteur de base de données Aurora MySQL 25/07/2023 (version 2.12.0, compatible avec MySQL 5.7.40) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 2024-07-19 (version 2.11.6, compatible avec MySQL 5.7.12) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 2024-03-26 (version 2.11.5, compatible avec MySQL 5.7.12) - Version RDS Extended Support (par défaut)
- Mises à jour du moteur de base de données Aurora MySQL 17/10/2023 (version 2.11.4, compatible avec MySQL 5.7.12) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 09/06/2023 (version 2.11.3, compatible avec MySQL 5.7.12) - Version RDS Extended Support

- Mises à jour du moteur de base de données Aurora MySQL 24/24 (version 2.11.2, compatible avec MySQL 5.7.12) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL 14/02/2023 (version 2.11.1, compatible avec MySQL 5.7.12) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL : 10-25 (version 2.11.0, compatible avec MySQL 5.7.12) - Version RDS Extended Support
- Mises à jour du moteur de base de données Aurora MySQL du 01/11/2021 (version 2.10.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 21/01/2022 (version 2.10.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 21/10/2021 (version 2.10.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 25/05/2021 (version 2.10.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 12/11/2022 (version 2.09.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 26/02/2021 (version 2.09.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 11/12/2020 (version 2.09.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 17/09/2020 (version 2.09.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 06/01/2022 (version 2.08.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 12/11/2022 (version 2.08.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 28/08/2020 (version 2.08.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 18/06/2020 (version 2.08.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 06/02/2020 (version 2.08.0) (obsolète)

- Mises à jour du moteur de base de données Aurora MySQL 15/08/15 (version 2.07.10) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL 04/03/2023 (version 2.07.9) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 16/06/2022 (version 2.07.8) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 24/11/2021 (version 2.07.7) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 02/09/2021 (version 2.07.6) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 06/07/2021 (version 2.07.5) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 04/03/2021 (version 2.07.) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 10/11/2020 (version 2.07.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 17/04/2020 (version 2.07.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 23/12/2019 (version 2.07.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 2.07.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 22/11/2019 (version 2.06.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 11/11/2019 (version 2.05.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 14/08/2020 (version 2.04.9) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 20/11/2019 (version 2.04.8) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 14/11/2019 (version 2.04.7) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 19/09/2019 (version 2.04.6) (obsolète)

- Mises à jour du moteur de base de données Aurora MySQL du 08/07/2019 (version 2.04.5) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 29/05/2019 (version 2.04.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 09/05/2019 (version 2.04.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 02/05/2019 (version 2.04.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 25/03/2019 (version 2.04.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 25/03/2019 (version 2.04.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 07/02/2019 (version 2.03.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 18/01/2019 (version 2.03.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 09/01/2019 (version 2.03.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 24/10/2018 (version 2.03.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 11/10/2018 (version 2.03) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 08/10/2018 (version 2.02.5) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 21/09/2018 (version 2.02.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 23/08/2018 (version 2.02.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 04/06/2018 (version 2.02.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 03/05/2018 (version 2.02) (obsolète)

- Mises à jour du moteur de base de données Aurora MySQL du 13/03/2018 (version 2.01.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 06/02/2018 (version 2.01) (obsolète)

# Mises à jour du moteur de base de données Aurora MySQL 2025-04-09 (version 2.12.5, compatible avec MySQL 5.7.44) - Version RDS Extended Support

Version : 2.12.5

Aurora MySQL 2.12.5 est généralement disponible. Les versions 2.12 d'Aurora MySQL sont compatibles jusqu'à MySQL 5.7.44. Pour plus d'informations sur les modifications apportées à la communauté, consultez les modifications apportées à MySQL 5.7.44 (11 octobre, disponibilité générale).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* et 3.08*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* existant vers Aurora MySQL 2.12.5. Vous pouvez également restaurer un instantané à partir de n'importe quelle version d'Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.12.5.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs :

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 5.7.44. Les correctifs CVE suivants sont inclus :

- CVE-2025-21543
- CVE-2025-21500
- CVE-2025-21491
- CVE-2025-21490
- CVE-2025-21540
- CVE-2025-21559
- CVE-2025-21555
- CVE-2025-21497
- CVE-2025-21520
- CVE-2025-21501
- CVE-2024-37371
- CVE-2024-11053
- CVE-2024-21201
- CVE-2024-21241
- CVE-2024-21230
- CVE-2023-44487

Améliorations de la disponibilité :

- Correction d'un problème sur la réplique à cause duquel une interruption du réseau risquait de ne pas rétablir correctement la connexion avec le rédacteur.

# Mises à jour du moteur de base de données Aurora MySQL 2024-10-23 (version 2.12.4, compatible avec MySQL 5.7.44) - Version RDS Extended Support

Version : 2.12.4

Aurora MySQL 2.12.4 est généralement disponible. Les versions 2.12 d'Aurora MySQL sont compatibles jusqu'à MySQL 5.7.44. Pour plus d'informations sur les modifications apportées à la

communauté, consultez les modifications apportées à MySQL 5.7.44 (11 octobre, disponibilité générale).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.* et 3.07.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* existant vers Aurora MySQL 2.12.4. Vous pouvez également restaurer un instantané à partir de n'importe quelle version d'Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.12.4.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité résolus et CVEs :

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 5.7.44. Les correctifs CVE suivants sont inclus :

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

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 5.7.44 incluse. Pour de plus amples informations, veuillez consulter Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lots des analyses

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Instruction SQL `CREATE TABLESPACE`
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL 2024-07-09 (version 2.12.3, compatible avec MySQL 5.7.44) - Version RDS Extended Support

Version : 2.12.3

Aurora MySQL 2.12.3 est généralement disponible. Les versions 2.12 d'Aurora MySQL sont compatibles jusqu'à MySQL 5.7.44. Pour plus d'informations sur les modifications apportées à la communauté, consultez les modifications apportées à MySQL 5.7.44 (11 octobre, disponibilité générale).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* et 3.07.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* existant vers Aurora MySQL 2.12.3. Vous pouvez également restaurer un instantané à partir de n'importe quelle version d'Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.12.3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité résolus et CVEs :

- Correction d'un problème de sécurité pour les procédures stockées MySQL.

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 5.7.44. Les correctifs CVE suivants sont inclus :

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

- CVE-2024-20998
- CVE-2024-21008
- CVE-2024-21009
- CVE-2024-21013
- CVE-2024-21047
- CVE-2024-21054
- CVE-2024-21055
- CVE-2024-21057
- CVE-2024-21062
- CVE-2024-21069
- CVE-2024-21096
- CVE-2024-21097

Améliorations de la disponibilité :

- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données Aurora MySQL lors de l'exécution d'une requête parallèle.
- Correction d'un problème qui provoquait le redémarrage d'un serveur de base de données en raison de l'accès simultané aux ressources de connexion lors d'une mise à l'échelle fluide, d'un redémarrage sans interruption (ZDR) et de l'application de correctifs sans interruption de service (ZDP).
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur lors de la libération de la mémoire utilisée pour l'application de journalisation.
- Correction d'un problème dans un processus en arrière-plan qui provoquait l'exécution prolongée ou l'échec des requêtes alors que l'opération en arrière-plan supprimait les index temporaires.
- Correction d'un problème lié aux routines de démarrage qui pouvaient provoquer le redémarrage des instances de base de données du rédacteur en raison d'une incohérence des métadonnées.
- Ajout d'un indicateur de la progression du recouvrement des transactions. Cela permet d'éviter une éventuelle indisponibilité, dans de rares situations, lorsque le rétablissement de la transaction prend beaucoup de temps.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur lors de la lecture d'une table modifiée ou supprimée sur l'instance de base de données d'écriture.

- Correction d'un problème à cause duquel une faible valeur de `thread_stack` paramètre provoquait le redémarrage répété de la base de données. La `thread_stack` valeur minimale autorisée a été augmentée de 131 072 à 136 192 pour garantir un démarrage réussi et éviter les problèmes de démarrage.
- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données de lecteur lors de l'exécution d'une requête parallèle.
- Correction d'un problème qui pouvait entraîner le redémarrage des répliques de lecture Aurora en cas de certains rares ordres de validation de transactions sur l'instance de base de données du rédacteur.
- Correction d'un problème qui, dans de rares cas, pouvait provoquer le redémarrage d'une instance de base de données lorsqu'une transaction en lecture seule obtenait des verrous partagés.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur utilisant le transfert d'écriture lorsqu'une instruction de validation implicite transférée rencontrait une erreur.

Améliorations générales :

- Correction d'un problème qui pouvait provoquer des erreurs inattendues ou des avertissements de violation de clé primaire sur certaines lignes lors de l'exécution d'`INSERT`instructions simultanées sur une table comportant une colonne de clé `AUTO_INCREMENT` primaire et une colonne de clé unique, et lorsqu'une `INSERT` instruction présentait des violations de clé uniques sur différentes lignes.
- Correction d'un problème qui pouvait entraîner des résultats de requête incorrects lorsque ZDR restaure de manière incorrecte les variables de session définies comme indices dans les requêtes.
- Correction d'un problème de requête parallèle qui provoquait le renvoi d'un jeu de résultats incomplet lors de l'utilisation des fonctions intégrées `LPAD` et des fonctions de `RPAD` chaîne.
- Correction d'un problème qui provoquait l'absence d'index de clé étrangère sur les instances de base de données du lecteur lorsqu'une `ALTER TABLE RENAME COLUMN` instruction était exécutée sur l'instance de base de données du rédacteur sur une table contenant une clé étrangère.
- Correction d'un problème pouvant entraîner l'échec du processus de désactivation du transfert d'écriture.
- Correction d'un problème lors du Aurora Serverless v1 dimensionnement qui provoquait le redémarrage de l'instance de base de données en raison d'un accès incorrect à une structure de données interne lors de la recherche d'un point de dimensionnement.

- Correction d'un problème en raison duquel le schéma de performance n'était pas activé lorsque la gestion automatisée de Performance Insights était activée pour les instances de base de données db.t4g.medium et db.t4g.large.
- Le délai d'expiration des demandes pour les opérations d'apprentissage automatique Aurora adressées à Amazon SageMaker AI a été augmenté de 3 à 30 secondes. Cela permet de résoudre un problème selon lequel les clients peuvent constater une augmentation du nombre de tentatives ou d'échecs pour les demandes adressées à Amazon SageMaker AI par le biais de l'apprentissage automatique Aurora lorsqu'ils utilisent des lots de plus grande taille.
- Correction d'un problème en raison duquel les `UPDATE` requêtes lentes `INSERT` et exécutées par le planificateur d'événements MySQL n'étaient pas enregistrées dans le journal des requêtes lentes sauf si elles étaient précédées d'une requête lente`SELECT`. `DELETE`

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 5.7.44 incluse. Pour de plus amples informations, veuillez consulter Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

- Correction d'un problème en raison duquel les tables temporaires liées à des déclencheurs lors de l'exécution d'instructions pouvaient provoquer un redémarrage inattendu du moteur de base de données.
- Correction d'un défaut qui pouvait entraîner la fermeture du serveur lorsque des instructions à table unique `UPDATE` et `DELETE` des instructions utilisant des expressions indexées étaient exécutées en tant qu'instructions préparées. (Insecte #29257254)

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lots des analyses

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL

utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Instruction SQL `CREATE TABLESPACE`
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL 2024-03-19 (version 2.12.2, compatible avec MySQL 5.7.44) - Version RDS Extended Support

Version : 2.12.2

Aurora MySQL 2.12.2 est généralement disponible. Les versions 2.12 d'Aurora MySQL sont compatibles jusqu'à MySQL 5.7.44. Pour plus d'informations sur les modifications apportées à la communauté, consultez les modifications apportées à MySQL 5.7.44 (11 octobre, disponibilité générale).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* et 3.06.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* existant vers Aurora MySQL 2.12.2. Vous pouvez également restaurer un instantané à partir de n'importe quelle version d'Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.12.2.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 5.7.44. Les correctifs CVE suivants sont inclus :

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

Problèmes de sécurité :

- Ajout d'un correctif qui garantit que les répliques de journaux binaires sont utilisées par défaut SSL/TLS si la source prend en charge les connexions chiffrées, quel que soit le `MASTER_SSL` paramètre.

Améliorations de la disponibilité :

- Correction d'un problème qui empêchait le lancement réussi d'une instance de réplication en lecture en cas de charge de travail élevée sur l'instance d'écriture.
- Correction d'un problème qui pouvait provoquer le basculement d'une instance de rédacteur de base de données Aurora MySQL en raison d'un défaut du composant qui communique avec le

stockage Aurora. Le défaut est dû à une rupture de communication entre l'instance de base de données et le stockage sous-jacent suite à une mise à jour logicielle de l'instance de stockage Aurora.

- Correction d'un problème qui, dans de rares cas, pouvait entraîner le redémarrage des instances du lecteur.
- Correction d'un problème qui empêchait un utilisateur privilégié de modifier les limites de ressources associées à l'utilisateur rdsadmin. Lorsqu'elles sont mal définies, ces limites de ressources peuvent empêcher l'agent de surveillance RDS de surveiller l'état de santé de l'instance de base de données, ce qui entraîne une indisponibilité de la base de données.

Mises à niveau et migrations :

- Correction d'un problème qui se produisait lors de la tentative de démarrage de la réplication des journaux binaires pour les clusters Aurora MySQL qui avaient été migrés depuis Amazon RDS MySQL 5.7 et qui contenaient des procédures stockées non prises en charge.
- Le planificateur d'événements de base de données a été désactivé lors d'une mise à niveau d'une version majeure vers Aurora MySQL version 3. Cette mise à jour permet d'éviter toute modification de la base de données par l'exécution d'événements alors que la mise à niveau de la version majeure est en cours.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 5.7.44 incluse. Pour plus d'informations, consultez Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lots des analyses

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Instruction SQL `CREATE TABLESPACE`
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL 28/12/2023 (version 2.12.1, compatible avec MySQL 5.7.40) - Version RDS Extended Support

Version : 2.12.1

Aurora MySQL 2.12.1 est en disponibilité générale. Les versions 2.12 d'Aurora MySQL sont compatibles avec MySQL jusqu'à 5.7.40. Pour plus d'informations sur les modifications, consultez Changes in MySQL 5.7.40 (2022-10-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* et 3.05.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.12.1. Vous pouvez également restaurer un instantané dans Aurora MySQL 2.12.1 à partir d'une version actuellement prise en charge d'Aurora MySQL.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Cette version inclut tous les CVEs correctifs communautaires, y compris MySQL 5.7.44.

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs de CVE supplémentaires ci-dessous :

- 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
- Correction du traitement des jetons à caractère unique par un plug-in d'analyse de recherche en texte intégral (bogue n° 35432973)
- Correction d'un problème en raison duquel les événements signalés lors du traitement des rotations du journal d'audit pouvaient ne pas être écrits dans le journal d'audit

Nouvelles fonctions :

- Ajout de la prise en charge de la réplication de journal binaire (binlog) multithread : le thread SQL au niveau du réplica du journal binaire peut ainsi appliquer les événements du journal binaire en parallèle lorsque cela est possible. Découvrez plus en détail les options de configuration permettant de peaufiner la réplication multithread dans le Guide de l'utilisateur Aurora.

Améliorations de la disponibilité :

- Correction d'un problème de redémarrage des instances de base de données Aurora MySQL utilisant une requête parallèle lorsqu'elles exécutent un grand nombre de requêtes parallèles simultanées.
- Correction d'un problème de conflit de verrou causé par un thread de journalisation d'audit pouvant conduire à une utilisation élevée du processeur et à des dépassements de délai d'attente pour les applications clientes.
- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données lors d'une tentative de lecture d'une page de base de données appartenant à une table supprimée.
- Correction d'un problème susceptible d'entraîner le redémarrage de l'instance d'enregistreur lorsque celle-ci augmente le volume de la base de données jusqu'à un multiple de 160 Go.
- Correction d'un problème dans le gestionnaire de verrous qui pouvait entraîner un redémarrage ou un basculement pendant le traitement de validations en deux phases, tandis que le niveau d'isolation était défini sur `READ_COMMITED` ou `READ_UNCOMMITED` et que des transactions XA étaient utilisées ou le journal binaire (binlog) activé.
- Correction d'un problème susceptible d'entraîner l'indisponibilité du cluster de base de données si l'instance d'enregistreur redémarre alors que la base de données crée ou supprime des déclencheurs sur des tables système internes.
- Correction d'un problème qui pouvait entraîner le redémarrage de l'instance de base de données lorsque le nombre de connexions à la base de données se rapprochait de la valeur définie par le paramètre `max_connections`.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de lecteur Aurora lors de l'exécution de requêtes DML (Data Manipulation Language) sur une table contenant un index en texte intégral.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT`

`INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème qui pouvait entraîner l'échec d'une requête parallèle en raison de problèmes réseau transitoires pendant la lecture de données issues du volume du cluster Aurora.
- Correction d'un problème lié à la gestion des fichiers journaux d'audit qui pouvait rendre les fichiers journaux inaccessibles au téléchargement ou à la rotation et qui, dans certains cas, faisait croître l'utilisation du processeur.
- Correction d'un problème d'augmentation du retard de réplication pour les petites instances de réplica en lecture après une mise à niveau à partir de versions antérieures à la version 2.11.*
- Correction d'un problème qui pouvait provoquer un nombre excessif de messages de journal lors de la consultation de table d'octrois `procs_priv` pour la vérification de requêtes impliquant des routines stockées.
- Correction d'un problème de gestion de la mémoire pouvant conduire l'instance de base de données à utiliser excessivement la mémoire lors de l'exécution de requêtes utilisant l'optimisation de la jointure de hachage.
- Correction d'un problème qui pouvait produire une valeur incorrecte pour la variable `Threads_running` dans les tables de statut global `information_schema` et `performance_schema` lors de l'utilisation du transfert d'écriture.
- Correction d'un problème provoquant le redémarrage de la base de données lors de l'exécution d'instructions `SELECT` avec des tables partitionnées (créées dans une version de MySQL prenant en charge l'ancien gestionnaire de partition `ha_partition`) et lorsque la requête parallèle est choisie par le planificateur de requêtes.
- Correction d'un problème empêchant l'établissement de nouvelles connexions clientes à la base de données lorsque le transfert d'écriture est activé.
- Réduction du retard de réplication du journal binaire lorsqu'un réplica du journal binaire Aurora MySQL exécute des événements `QUERY` écrits dans le fichier binaire de la source sans qu'une base de données par défaut soit définie par la commande `USE`.
- Correction d'un problème qui pouvait entraîner l'`CommitLatency` CloudWatch affichage incorrect de la métrique lorsque le `innodb_flush_log_at_trx_commit` paramètre n'était pas défini sur 1.

- Correction d'un problème pouvant entraîner l'arrêt des connexions aux bases de données avant qu'elles ne soient établies. Ce problème est plus susceptible d'affecter les instances de bases de données qui ouvrent et ferment beaucoup de connexions.
- Correction d'un problème qui pouvait provoquer le redémarrage de la base de données lorsque les consommateurs de journaux binaires (binlog) connectés utilisaient un serveur de réplication de journaux binaires dupliqué. IDs
- Correction d'un problème qui pouvait entraîner l'arrêt de la réplication des journaux binaires multithread lorsque la `replica_parallel_type` variable était définie sur `LOGICAL_CLOCK` et que la `replica_preserve_commit_order` variable était activée. `ON` Ce problème peut se produire lorsqu'une transaction supérieure à 500 Mo est exécutée sur la source.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 5.7.40 incluse. Pour plus d'informations, consultez Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

- Correction d'un problème qui pouvait entraîner le blocage des connexions à distance existantes et nouvelles lorsqu'elles étaient exécutées simultanément avec l'instruction `SHOW PROCESSLIST` (bogue Community n° 34857411)
- Réplication : certains événements de journal binaire n'étaient pas toujours gérés correctement (bogue n° 34617506).

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lots des analyses

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des

performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Instruction SQL `CREATE TABLESPACE`
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL 25/10/2023 (version 2.12.0.1, compatible avec MySQL 5.7.40) - Version RDS Extended Support (bêta)

Version : 2.12.0.1

Aurora MySQL 2.12.0.1 est généralement disponible dans les régions suivantes : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Californie du Nord), USA Ouest (Oregon), (USA Est) et AWS GovCloud AWS GovCloud (US-Ouest). Il s'agit d'une version anticipée qui ne contient que des correctifs de sécurité. Ces correctifs seront déployés plus largement dans toutes les régions avec le prochain correctif (version 2.12.1). Les versions 2.12 d'Aurora MySQL sont compatibles avec MySQL 5.7.40.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, et 3.04.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* existant vers Aurora MySQL 2.12.0.1. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.12.0.1.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Cette version inclut tous les CVEs correctifs communautaires, y compris MySQL 5.7.40.

- CVE-2023-38545

Améliorations de la disponibilité :

- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

## Mises à jour du moteur de base de données Aurora MySQL 25/07/2023 (version 2.12.0, compatible avec MySQL 5.7.40) - Version RDS Extended Support

Version : 2.12.0

Aurora MySQL 2.12.0 est disponible. Les versions 2.12 d'Aurora MySQL sont compatibles avec MySQL jusqu'à 5.7.40. Pour plus d'informations sur les modifications, consultez Changes in MySQL 5.7.40 (2022-10-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont 2.11.*, 2.12.*, 3.01.*, 3.02.* et 3.03.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* vers Aurora MySQL 2.12.0. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.12.0.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Cette version inclut tous les CVEs correctifs communautaires, y compris MySQL 5.7.40.

- Les chiffrements SSL par défaut utilisés par Aurora MySQL ont été mis à jour pour exclure les valeurs DES- CBC3 -SHA moins sécurisées du paramètre de base de données SSL_CIPHER. Si vous rencontrez des problèmes de connexion SSL en raison de la suppression du chiffrement DES- CBC3 -SHA, veuillez utiliser un algorithme de chiffrement sécurisé applicable figurant dans la liste suivante, Configuration des suites de chiffrement pour les connexions aux clusters de bases de données Aurora MySQL. Vous trouverez plus d'informations sur le client MySQL Connection Cipher Configuration dans la documentation MySQL.
- CVE-2023-21963
- CVE-2023-21912
- CVE-2023-21840
- CVE-2023-0215
- CVE-2022-43551
- CVE-2022-37434
- CVE-2022-32221
- CVE-2021-36222

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

Améliorations de la disponibilité :

- Correction d'un problème de chiffrement des événements DAS (Database Activity Streams) pouvant entraîner le redémarrage de la base de données.
- Correction de deux problèmes pouvant faire échouer le redémarrage d'une base de données si celui-ci se produisait lors de l'exécution d'une requête DDL (Data Definition Language).
- Correction d'un problème de surcroît de connexions qui peut entraîner une augmentation de la latence des requêtes ou un redémarrage de l'instance de base de données.
- Correction d'un problème susceptible, dans de rares cas, de provoquer le redémarrage d'un réplica Aurora lors de l'exécution simultanée d'opérations de mise à jour volumineuses ou de charges de travail DDL (Data Definition Language) sur l'instance d'enregistreur et d'opérations de lecture sur le même ensemble de tables sur le réplica Aurora.
- Correction d'un problème de surcroît de connexions qui peut entraîner un allongement de la durée du processus d'établissement de la connexion ou provoquer un échec en raison d'erreurs de délai d'expiration.
- Correction d'un problème lors duquel la rotation du journal d'audit avancé pouvait réduire la mémoire disponible, ce qui pouvait entraîner le redémarrage de l'instance de base de données.
- Correction d'un problème pouvant entraîner le redémarrage d'une instance de lecteur Aurora MySQL lors de l'exécution d'une requête utilisant un plan d'exécution de requêtes parallèles Aurora.
- Correction d'un problème qui provoquait le redémarrage de l'instance d'enregistreur lors de l'exécution de la requête `OPTIMIZE TABLE` sur une table dotée d'un index Full Text Search (FTS).
- Correction d'un problème qui provoquait le redémarrage de l'instance d'écriture d'une AWS région principale de base de données globale Aurora lorsqu'une `SELECT FOR UPDATE` requête était exécutée à l'aide du transfert d'écriture global depuis une région secondaire de base de données globale Aurora.
- Correction d'un problème qui provoquait le redémarrage d'une instance de lecteur de AWS région secondaire de la base de données globale Aurora utilisant le transfert d'écriture global lorsqu'une instruction de validation implicite transférée rencontrait une erreur.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT`

`INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Introduction d'optimisations des performances de gestion des fichiers sur les réplicas des fichiers binaires afin de réduire les conflits lors de l'écriture dans les fichiers journaux de relais.
- Correction d'un problème pouvant entraîner un signalement incorrect du compteur `buffer_pool_read_requests` dans les métriques `information_schema`.
- Correction d'un problème pouvant entraîner le remplissage de l'espace de stockage local lors de l'exécution d'opérations `LOAD FROM S3` ou `SELECT INTO S3`. Le problème peut également entraîner une augmentation de l'utilisation du processeur, des redémarrages de la base de données en raison d'une mémoire insuffisante et une augmentation de la latence pour ces requêtes.
- Correction d'un problème lors duquel des instances de base de données utilisant la réplication de journaux binaires pouvaient connaître une augmentation de l'utilisation du processeur et des échecs de connexion lorsque plusieurs consommateurs de la réplication de journaux binaires étaient connectés.
- Correction d'un problème lors duquel les variables d'état du serveur SSL n'étaient pas renseignées.
- Correction d'un problème lors duquel des instructions DML (Data Manipulation Language) exécutant des écritures en double pouvaient entraîner une journalisation excessive des erreurs et une augmentation de la latence des requêtes.
- Mise à niveau des définitions de fuseaux horaires vers la version IANA 2023c.
- Ajout de la prise en charge de l'activation et de la désactivation de la journalisation binaire au niveau de la session. Consultez la section Procédures stockées - Réplication dans le Guide de l'utilisateur Amazon Aurora.
- Ajout de la prise en charge de la définition du format de journal binaire au niveau de la session. Consultez la section Procédures stockées - Réplication dans le Guide de l'utilisateur Amazon Aurora.
- Correction d'un problème lors duquel le réglage du paramètre `aurora_disable_hash_join` sur `1` ou `ON` n'empêchait pas l'optimiseur d'utiliser une jointure de hachage.
- Correction d'un problème avec l'analyse d'index qui pouvait renvoyer un résultat inexact lors de l'exécution d'une requête `SELECT` avec la clause `GROUP BY` et le paramètre `aurora_parallel_query` défini sur `ON`.

- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage d'une instance Amazon Aurora lors de l'accès à une table comportant une mise à jour importante ou des opérations DDL (Data Definition Language) exécutées simultanément sur l'instance d'enregistreur.
- Correction d'un problème pouvant entraîner un signalement incorrect du compteur `buffer_pool_read_requests` dans les métriques `information_schema`.
- Correction d'un problème provoquant le redémarrage d'un réplica du journal binaire si la variable système server_uuid de la source est absente ou a une valeur non valide.
- Correction d'un problème empêchant l'obsolescence des statistiques d'InnoDB, ce qui pouvait parfois générer un plan d'exécution de requêtes sous-optimal susceptible d'entraîner une augmentation du temps d'exécution des requêtes.
- Correction d'un problème à cause duquel les `AuroraGlobalDBRPOLag` CloudWatch métriques affichaient toujours zéro, quelle que soit la charge de travail de l'utilisateur

Mises à niveau et migrations :

- Pour effectuer la mise à niveau d'une version mineure d'une base de données globale Aurora MySQL version 32.07 ou 2.11 vers Aurora MySQL version 2.12 ou ultérieure, consultez Mise à niveau d'Aurora MySQL par modification de la version du moteur.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 5.7.40 incluse. Pour plus d'informations, consultez Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

- Correction d'un problème susceptible d'entraîner une utilisation plus élevée du processeur en raison de la rotation des certificats TLS en arrière-plan. (Bogue n° 34284186)

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lot des analyses.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Instruction SQL `CREATE TABLESPACE`
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du groupe de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL 2024-07-19 (version 2.11.6, compatible avec MySQL 5.7.12) - Version RDS Extended Support

Version : 2.11.6

Aurora MySQL 2.11.6 est généralement disponible. Les versions 2.11 d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* et 3.07.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* existant vers Aurora MySQL 2.11.6. Vous pouvez également restaurer un instantané à partir de n'importe quelle version inférieure d'Aurora MySQL version 2 actuellement prise en charge dans Aurora MySQL 2.11.6.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 2.11.*, vous devez mettre à niveau vos clusters de bases de données principal et secondaire vers la même version, y compris le niveau de correctif. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Immédiatement après l'exécution d'une mise à niveau sur place de la version du moteur vers Aurora MySQL 2.11.*, une mise à niveau du système d'exploitation est appliquée automatiquement à toutes les instances concernées sur les classes d'instances de base de données db.r4, db.r5, db.t2 et db.t3, si les instances exécutent une ancienne version du système d'exploitation. Dans un cluster de base de données multi-AZ, toutes les instances de lecteur appliquent d'abord la mise à niveau du système d'exploitation. Lorsque la mise à niveau du système d'exploitation sur la première instance d'enregistreur est terminée, un basculement se produit et l'instance d'enregistreur précédente est mise à niveau.

> **Note**
>
> La mise à niveau du système d'exploitation n'est pas appliquée automatiquement aux bases de données globales Aurora lors des mises à niveau majeures des versions.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité résolus et CVEs :

- Correction d'un problème de sécurité dans les procédures stockées MySQL.

Cette version inclut tous les correctifs CVE communautaires, y compris MySQL 5.7.12. Les correctifs CVE suivants sont inclus dans cette version :

- CVE-2023-21912
- CVE-2023-38545
- CVE-2023-39975
- CVE-2023-44487
- CVE-2024-0853
- CVE-2024-20963
- CVE-2024-20993
- CVE-2024-20998
- CVE-2024-21008
- CVE-2024-21009
- CVE-2024-21013
- CVE-2024-21047
- CVE-2024-21054
- CVE-2024-21055
- CVE-2024-21057
- CVE-2024-21062
- CVE-2024-21069
- CVE-2024-21096
- CVE-2024-21097

Améliorations de la disponibilité :

- Correction d'un problème qui pouvait provoquer le redémarrage d'un serveur de base de données en raison de l'accès simultané aux ressources de connexion lors d'une mise à l'échelle fluide, d'un redémarrage sans interruption (ZDR) et d'un correctif sans interruption de service (ZDP).
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur lors de la libération de la mémoire utilisée pour l'application de journalisation.

- Correction d'un problème qui provoquait le redémarrage d'une instance de base de données de lecteur lors de l'exécution d'une requête parallèle.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données de lecteur utilisant le transfert d'écriture lorsqu'une instruction de validation implicite transférée rencontrait une erreur.

Améliorations générales :

- Correction d'un problème qui pouvait provoquer des erreurs inattendues ou des avertissements de violation de clé primaire sur certaines lignes lors de l'exécution d'`INSERT`instructions simultanées sur une table comportant une colonne de clé `AUTO_INCREMENT` primaire et une colonne de clé unique, et lorsqu'une `INSERT` instruction présentait des violations de clé uniques sur différentes lignes.
- Correction d'un problème qui pouvait entraîner des résultats de requête incorrects lorsque ZDR restaure de manière incorrecte les variables de session définies comme indices dans les requêtes.
- Correction d'un problème lors du Aurora Serverless v1 dimensionnement qui provoquait le redémarrage de l'instance de base de données en raison d'un accès incorrect à une structure de données interne lors de la recherche d'un point de dimensionnement.
- Correction d'un problème en raison duquel les `UPDATE` requêtes lentes `INSERT` et exécutées par le planificateur d'événements MySQL n'étaient pas enregistrées dans le journal des requêtes lentes sauf si elles étaient précédées d'une requête lente`SELECT`. `DELETE`

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Cette version inclut toutes les corrections de bogues communautaires jusqu'à la version 5.7.12 incluse. Pour de plus amples informations, veuillez consulter Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lot des analyses. Pour de plus amples informations, veuillez consulter Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL 2024-03-26 (version 2.11.5, compatible avec MySQL 5.7.12) - Version RDS Extended Support (par défaut)

Version : 2.11.5

Aurora MySQL 2.11.5 est généralement disponible. Les versions 2.11 d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

La version 2.11.5 est la version par défaut actuelle d'Aurora MySQL version 2 lorsque vous créez un cluster de base de données. Pour plus d'informations, consultez la section Versions par défaut d'Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.*, 3.05.* et 3.06.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* existant vers Aurora MySQL 2.11.5. Vous pouvez également restaurer un instantané à partir de n'importe quelle version inférieure d'Aurora MySQL version 2 actuellement prise en charge dans Aurora MySQL 2.11.5.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 2.11.*, vous devez mettre à niveau vos clusters de bases de données principal et secondaire vers la même version, y compris le niveau de correctif. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Immédiatement après l'exécution d'une mise à niveau sur place de la version du moteur vers Aurora MySQL 2.11.*, une mise à niveau du système d'exploitation est appliquée automatiquement à toutes les instances concernées sur les classes d'instances de base de données db.r4, db.r5, db.t2 et db.t3, si les instances exécutent une ancienne version du système d'exploitation. Dans un cluster de base de données multi-AZ, toutes les instances de lecteur appliquent d'abord la mise à niveau du système d'exploitation. Lorsque la mise à niveau du système d'exploitation sur la première instance d'enregistreur est terminée, un basculement se produit et l'instance d'enregistreur précédente est mise à niveau.

> **Note**
>
> La mise à niveau du système d'exploitation n'est pas appliquée automatiquement aux bases de données globales Aurora lors des mises à niveau majeures des versions.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

# Améliorations

Problèmes de sécurité corrigés et CVEs :

Les correctifs CVE suivants sont inclus dans cette version :

- 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

Améliorations de la disponibilité :

- Correction d'un problème à cause duquel une instance de base de données Aurora MySQL Writer pouvait basculer en raison d'un défaut du composant qui communique avec le stockage Aurora. Le défaut est dû à une rupture de communication entre l'instance de base de données et le stockage sous-jacent suite à une mise à jour logicielle.
- Correction d'un problème qui, dans de rares cas, pouvait entraîner le redémarrage des instances de base de données du lecteur.
- Correction d'un problème de conflit de verrou causé par un thread de journalisation d'audit pouvant conduire à une utilisation élevée du processeur et à des dépassements de délai d'attente pour les applications clientes.

Améliorations générales :

- Correction d'un problème qui pouvait entraîner l'échec d'une requête parallèle en raison de problèmes réseau transitoires lors de la lecture des données du volume du cluster de base de données Aurora.

- Correction d'un problème lié à la gestion des fichiers journaux d'audit qui pouvait rendre les fichiers journaux inaccessibles pour le téléchargement ou la rotation et, dans certains cas, augmenter l'utilisation du processeur.
- Correction d'un problème qui pouvait générer une valeur incorrecte de la `Threads_running` variable dans les tables d'état `information_schema` et les tables d'état `performance_schema` globales lors de l'utilisation du transfert d'écriture.

Mises à niveau et migrations :

- Correction d'un problème qui empêchait le lancement de la réplication des journaux binaires sur les clusters de bases de données Aurora MySQL migrés depuis RDS for MySQL 5.7.
- Le planificateur d'événements de base de données a été désactivé lors des mises à niveau majeures vers Aurora MySQL version 3. Cela permet d'éviter toute modification de la base de données par l'exécution d'événements pendant la mise à niveau de la version majeure.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 5.7.12 incluse. Pour plus d'informations, consultez Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lot des analyses. Pour de plus amples informations, veuillez consulter Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL 17/10/2023 (version 2.11.4, compatible avec MySQL 5.7.12) - Version RDS Extended Support

Version : 2.11.4

Aurora MySQL 2.11.4 est disponible. Les versions 2.11 d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.* et 3.04.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* vers Aurora MySQL 2.11.4. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.11.4.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 2.11.*, vous devez mettre à niveau vos clusters de bases de données principal et secondaire vers la même version, y compris le niveau de correctif. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Immédiatement après l'exécution d'une mise à niveau sur place de la version du moteur vers Aurora MySQL 2.11.*, une mise à niveau du système d'exploitation est appliquée automatiquement à toutes les instances concernées sur les classes d'instances de base de données db.r4, db.r5, db.t2 et db.t3, si les instances exécutent une ancienne version du système d'exploitation. Dans un cluster de base de données multi-AZ, toutes les instances de lecteur appliquent d'abord la mise à niveau du système d'exploitation. Lorsque la mise à niveau du système d'exploitation sur la première instance d'enregistreur est terminée, un basculement se produit et l'instance d'enregistreur précédente est mise à niveau.

> **Note**
>
> La mise à niveau du système d'exploitation n'est pas appliquée automatiquement aux bases de données globales Aurora lors des mises à niveau majeures des versions.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

- Correction d'un problème susceptible d'empêcher l'écriture dans le journal d'audit d'événements signalés lors du traitement des rotations du journal d'audit.
- CVE-2022-24407

Améliorations de la disponibilité :

- Correction d'un problème de redémarrage des instances de base de données Aurora MySQL utilisant une requête parallèle lorsqu'elles exécutent un grand nombre de requêtes parallèles simultanées.
- Correction d'un problème qui pouvait provoquer le redémarrage d'une instance de base de données lors de l'exécution de charges de travail de lecture I/O intensives.
- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données lors d'une tentative de lecture d'une page de base de données appartenant à une table supprimée.
- Correction d'un problème susceptible d'entraîner le redémarrage de l'instance d'enregistreur lorsque celle-ci augmente le volume de la base de données jusqu'à un multiple de 160 Go.
- Correction d'un problème susceptible d'entraîner l'indisponibilité du cluster de base de données si l'instance d'enregistreur redémarre alors que la base de données crée ou supprime des déclencheurs sur des tables système internes.
- Correction d'un problème pouvant provoquer le redémarrage d'une instance de lecteur lors de l'exécution de requêtes DML (Data Manipulation Language) sur une table contenant un index Full Text.
- Correction d'un problème pouvant entraîner le redémarrage d'une instance de lecteur lors de l'exécution d'une requête utilisant un plan d'exécution de requêtes Aurora parallèle.
- Correction d'un problème pouvant entraîner le redémarrage de l'instance d'enregistreur lors de l'exécution de la requête `OPTIMIZE TABLE` sur une table dotée d'un index Full Text Search (FTS).
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème d'augmentation du retard de réplication pour les petites instances de réplica en lecture après une mise à niveau à partir de versions antérieures à la version 2.11.*.
- Correction d'un problème pouvant provoquer un nombre excessif de messages de journaux lors de la consultation de table des autorisations procs_priv pour la vérification de requêtes impliquant des routines stockées.

- Correction d'un problème de gestion de la mémoire pouvant conduire l'instance de base de données à utiliser excessivement la mémoire lors de l'exécution de requêtes utilisant l'optimisation de la jointure de hachage.
- Correction d'un problème provoquant le redémarrage de la base de données lors de l'exécution d'instructions `SELECT` avec des tables partitionnées (créées dans une version de MySQL prenant en charge l'ancien gestionnaire de partition `ha_partition`) et lorsque la requête parallèle est choisie par le planificateur de requêtes.
- Correction d'un problème empêchant l'établissement de nouvelles connexions clientes à la base de données lorsque le transfert d'écriture est activé.
- Réduction du retard de réplication du journal binaire lorsqu'un réplica du journal binaire Aurora MySQL exécute des événements `QUERY` écrits dans le fichier binaire de la source sans qu'une base de données par défaut soit définie par la commande `USE`.
- Correction d'un problème avec l'analyse d'index qui pouvait renvoyer un résultat inexact lors de l'exécution d'une requête `SELECT` avec la clause `GROUP BY` et le paramètre `aurora_parallel_query` défini sur `ON`.
- Ajout de la prise en charge de l'activation et de la désactivation de la journalisation binaire au niveau de la session. Consultez la section Procédures stockées - Réplication dans le Guide de l'utilisateur Amazon Aurora.
- Correction d'un problème provoquant le redémarrage d'un réplica du journal binaire si la variable système server_uuid de la source est absente ou a une valeur non valide.
- Ajout de la prise en charge de la définition du format du journal binaire au niveau de la session. Consultez la section Procédures stockées - Réplication dans le Guide de l'utilisateur Amazon Aurora.
- Correction d'un problème qui pouvait entraîner l'`CommitLatency` CloudWatch affichage incorrect de la métrique lorsque le `innodb_flush_log_at_trx_commit` paramètre n'était pas défini sur 1.
- Correction d'un problème empêchant l'obsolescence des statistiques d'InnoDB, ce qui pouvait parfois générer un plan d'exécution de requêtes sous-optimal qui pouvait entraîner une augmentation de la durée d'exécution des requêtes.
- Correction d'un problème qui pouvait provoquer le redémarrage de la base de données lorsque les consommateurs de journaux binaires (binlog) connectés utilisaient un serveur de réplication de journaux binaires dupliqué. IDs

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 5.7.12 incluse. Pour plus d'informations, consultez Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

- Réplication : certains événements du journal binaire n'étaient pas toujours gérés correctement. (Bogue n° 34617506)
- Correction d'un problème susceptible d'entraîner une augmentation de l'utilisation du processeur en raison de la rotation des certificats TLS en arrière-plan. (Bogue n° 34284186).
- Dans les instructions préparées, certains types de sous-requête pouvaient causer un arrêt du serveur. (Bogue n° 33100586)

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lot des analyses. Pour de plus amples informations, veuillez consulter Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB

- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL 09/06/2023 (version 2.11.3, compatible avec MySQL 5.7.12) - Version RDS Extended Support

Version : 2.11.3

Aurora MySQL 2.11.3 est disponible. Les versions 2.11 d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.*, 3.01.*, 3.02.* et 3.03.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* vers Aurora MySQL 2.11.3. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.11.3.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 2.11.*, vous devez mettre à niveau vos clusters de bases de données principal et secondaire vers la même version, y compris le niveau de correctif. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Immédiatement après l'exécution d'une mise à niveau sur place de la version du moteur vers Aurora MySQL 2.11.*, une mise à niveau du système d'exploitation est appliquée automatiquement à toutes les instances concernées sur les classes d'instances de base de données db.r4, db.r5, db.t2 et db.t3, si les instances exécutent une ancienne version du système d'exploitation. Dans un cluster de base de données multi-AZ, toutes les instances de lecteur appliquent d'abord la mise à niveau du système d'exploitation. Lorsque la mise à niveau du système d'exploitation sur la première instance

d'enregistreur est terminée, un basculement se produit et l'instance d'enregistreur précédente est mise à niveau.

> **Note**
>
> La mise à niveau du système d'exploitation n'est pas appliquée automatiquement aux bases de données globales Aurora lors des mises à niveau majeures des versions.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

- Mise à jour des chiffrements SSL par défaut utilisés par Aurora MySQL pour exclure les valeurs `DES-CBC3-SHA` les moins sécurisées du paramètre SSL_CIPHER de la de base de données. Si vous rencontrez des problèmes de connexion SSL en raison de la suppression du `DES-CBC3-SHA` chiffrement, veuillez utiliser un chiffrement sécurisé applicable de cette liste. ConfiguringCipherSuites Vous trouverez plus d'informations sur le client MySQL Connection Cipher Configuration dans la documentation MySQL.
- CVE-2023-21963
- CVE-2023-21912
- CVE-2023-0215
- CVE-2022-43551
- CVE-2022-37434

Améliorations de la disponibilité :

- Correction d'un problème de chiffrement des événements DAS (Database Activity Streams) pouvant entraîner le redémarrage de la base de données.
- Correction de deux problèmes pouvant faire échouer le redémarrage d'une base de données si celui-ci se produisait lors de l'exécution d'une requête DDL (Data Definition Language).
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Optimisation des performances de la gestion des fichiers sur les réplicas de fichiers binaires afin de réduire les conflits d'écriture dans les fichiers journaux de relais.
- Correction d'un problème lors duquel le réglage du paramètre `aurora_disable_hash_join` sur 1 ou ON n'empêchait pas l'optimiseur d'utiliser une jointure de hachage.
- Correction d'un problème pouvant entraîner un signalement incorrect du compteur `buffer_pool_read_requests` dans les métriques `information_schema`.
- Correction d'un problème pouvant entraîner le remplissage de l'espace de stockage local lors de l'exécution d'opérations `LOAD FROM S3` ou `SELECT INTO S3`. Le problème peut également entraîner une augmentation de l'utilisation du processeur, des redémarrages de la base de données en raison d'une mémoire insuffisante et une augmentation de la latence pour ces requêtes.

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lot des analyses. Pour de plus amples informations, veuillez consulter Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL

utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL 24/24 (version 2.11.2, compatible avec MySQL 5.7.12) - Version RDS Extended Support

Version : 2.11.2

Aurora MySQL 2.11.2 est disponible. Les versions 2.11 d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.*, 3.01.*, 3.02.* et 3.03.*.

Vous pouvez mettre à niveau un cluster de base de données Aurora MySQL 2.* vers Aurora MySQL 2.11.2. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.11.2.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 2.11.*, vous devez mettre à niveau vos clusters de bases de données principal et secondaire vers la même version, y compris le niveau de correctif. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Immédiatement après l'exécution d'une mise à niveau sur place de la version du moteur vers Aurora MySQL 2.11.*, une mise à niveau du système d'exploitation est appliquée automatiquement à toutes les instances concernées sur les classes d'instances de base de données db.r4, db.r5, db.t2 et db.t3, si les instances exécutent une ancienne version du système d'exploitation. Dans un cluster de base de données multi-AZ, toutes les instances de lecteur appliquent d'abord la mise à niveau du système d'exploitation. Lorsque la mise à niveau du système d'exploitation sur la première instance d'enregistreur est terminée, un basculement se produit et l'instance d'enregistreur précédente est mise à niveau.

> **Note**
>
> La mise à niveau du système d'exploitation n'est pas appliquée automatiquement aux bases de données globales Aurora lors des mises à niveau majeures des versions.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Améliorations générales :

- Correction d'un problème lors duquel des instances de base de données utilisant la réplication des journaux binaires pouvaient connaître une augmentation de l'utilisation du processeur et des échecs de connexion lorsque plusieurs consommateurs de la réplication de journaux binaires sont connectés.

- Correction d'un problème pouvant entraîner la désynchronisation d'une instance de lecteur dans une région secondaire de la base de données globale après la mise à niveau vers Aurora MySQL version 2.11 si l'enregistreur de la base de données principale est sur Aurora MySQL version 2.10.

Améliorations de la disponibilité :

- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Traitement par lot des analyses. Pour de plus amples informations, veuillez consulter Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne

- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL 14/02/2023 (version 2.11.1, compatible avec MySQL 5.7.12) - Version RDS Extended Support

Version : 2.11.1

Aurora MySQL 2.11.1 est disponible. Les versions 2.11 d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.07.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.11.1. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 ou version ultérieure directement vers la version 2.11.1. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.11.1.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 2.11.* et que le transfert d'écriture est activé, vous devez mettre à niveau vos clusters de bases de données principal et secondaire vers la même version, y compris le niveau de correctif, pour continuer à utiliser le transfert d'écriture. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Immédiatement après l'exécution d'une mise à niveau sur place de la version du moteur vers Aurora MySQL 2.11.*, une mise à niveau du système d'exploitation est appliquée automatiquement à toutes les instances concernées sur les classes d'instances de base de données db.r4, db.r5, db.t2 et db.t3, si les instances exécutent une ancienne version du système d'exploitation. Dans un cluster de base de données multi-AZ, toutes les instances de lecteur appliquent d'abord la mise à niveau du système d'exploitation. Lorsque la mise à niveau du système d'exploitation sur la première instance

d'enregistreur est terminée, un basculement se produit et l'instance d'enregistreur précédente est mise à niveau.

> **Note**
>
> La mise à niveau du système d'exploitation n'est pas appliquée automatiquement aux bases de données globales Aurora lors des mises à niveau majeures des versions.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

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

Améliorations de la disponibilité :

- Correction d'un problème de surcroît de connexions qui peut entraîner une augmentation de la latence des requêtes ou un redémarrage de l'instance de base de données.
- Correction d'un problème susceptible, dans de rares cas, de provoquer le redémarrage d'un réplica Aurora lors de l'exécution simultanée d'opérations de mise à jour volumineuses ou de charges de

travail DDL (Data Definition Language) sur l'instance d'enregistreur et d'opérations de lecture sur le même ensemble de tables sur le réplica Aurora.

- Correction d'un problème de surcroît de connexions qui peut entraîner un allongement de la durée du processus d'établissement de la connexion ou provoquer un échec en raison d'erreurs de délai d'expiration.
- Correction d'un problème de réduction de la mémoire disponible à cause de la rotation du journal d'audit avancé, qui pouvait entraîner le redémarrage de l'instance de base de données.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème d'absence de renseignement des variables d'état du serveur SSL.
- Correction d'un problème lors duquel des instructions DML (Data Manipulation Language) exécutant des écritures en double pouvaient entraîner une journalisation excessive des erreurs et une augmentation de la latence des requêtes.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL : 10-25 (version 2.11.0, compatible avec MySQL 5.7.12) - Version RDS Extended Support

Version : 2.11.0

Aurora MySQL 2.11.0 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.11.0. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 ou version ultérieure directement vers la version 2.11.0. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge par Aurora MySQL 2.11.0.

Si vous mettez à niveau une base de données globale Aurora MySQL vers la version 2.11.* et que le transfert d'écriture est activé, vous devez mettre à niveau vos clusters de bases de données principal et secondaire vers la même version, y compris le niveau de correctif, pour continuer à utiliser le transfert d'écriture. Pour plus d'informations sur la mise à niveau de la version mineure d'une base de données globale Aurora, consultez Mises à niveau de version mineure.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs :

- Le SSL/TLS fournisseur est passé d'AWS-LCOpenSSL à.

  Cela apporte un certain nombre de changements, y compris, mais sans s'y limiter, la suppression du support pour les chiffrements DHE-RSA-* moins sécurisés.

Pour plus d'informations, consultez Utilisation de TLS avec les clusters de bases de données Aurora MySQL

Les correctifs CVE suivants sont inclus dans cette version :

- 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

Nouvelles fonctions :

- Avec la sortie d'Aurora MySQL version 2.11, une nouvelle mise à niveau du système d'exploitation est disponible. Nous vous recommandons d'appliquer cette mise à jour du système d'exploitation en attente à toutes vos instances de base de données Aurora MySQL après avoir effectué la mise à niveau vers la version 2.11. Pour plus d'informations, consultez Utilisation des mises à jour du système d'exploitation.
- Une nouvelle option de configuration dynamique, innodb_deadlock_detect, peut être utilisée pour désactiver la détection des interblocages. Sur les systèmes à forte simultanéité, la détection des interblocages peut entraîner un ralentissement lorsque de nombreux threads attendent le même verrouillage. Parfois, il peut être plus efficace de désactiver la détection des interblocages et d'utiliser le paramètre innodb_lock_wait_timeout pour annuler des transactions en cas d'interblocage. (Bogue n° 23477773) Pour plus d'informations sur la détection des interblocages dans Innodb, consultez la documentation MySQL.
- Les fonctions `UUID_TO_BIN`, `BIN_TO_UUID` et `IS_UUID` de MySQL 8.0 ont été ajoutées. Pour plus d'informations sur l'utilisation de ces fonctions, consultez MySQL Miscellaneous functions.
- Ajout de la prise en charge des indicateurs d'optimisation permettant à l'utilisateur d'activer ou de désactiver une requête parallèle Aurora MySQL par table ou par requête.
  - Utilisation des requêtes parallèles pour Amazon Aurora MySQL
  - Indicateurs Aurora MySQL
- Suppression de la prise en charge des types d'instance R3.
- Ajout de la prise en charge des instances R6i.

Améliorations de la disponibilité :

- Correction d'un problème qui empêchait la réplication logique entre régions dans un cluster de base de données en raison d'un nom et d'une position de fichier binaire incorrects dans les journaux d'erreurs. Ce problème peut se produire lors du redémarrage du moteur après l'exécution d'une instruction DDL.
- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage d'instances de lecteur Aurora lors de l'exécution d'instructions de liste de contrôle d'accès (ACL) telles que GRANT et FLUSH sur l'instance d'enregistreur. Ce problème est plus susceptible d'affecter les instances de lecteur comprenant un grand nombre d'utilisateurs et les opérations ACL (modifications d'autorisations, par exemple).
- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage ou le basculement de l'instance d'enregistreur lorsqu'une transaction accède à une ligne supprimée par une autre transaction.

- Amélioration des performances de la recherche de phrases Fulltext afin de réduire de manière significative le temps nécessaire à la recherche de phrases dans une table contenant des index Fulltext.
- Correction d'un problème de récupération d'une durée excessive à la suite du redémarrage d'une instance d'enregistreur, suivie d'un nouveau redémarrage. Ce problème se produit lorsque la base de données comprend un grand nombre de lignes non validées au moment du premier redémarrage.
- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage du serveur de base de données en raison d'un long délai d'attente du sémaphore lorsque le thread de détecteur d'interblocages est bloqué.
- Correction d'un problème qui, dans de rares cas, pouvait entraîner le redémarrage de la base de données en raison d'une longue attente au sémaphore lorsque les I/O threads étaient bloqués.
- L'insertion rapide n'est pas activée dans cette version d'Aurora MySQL, en raison d'un problème susceptible de provoquer des incohérences lors de l'exécution de requêtes telles que `INSERT INTOSELECT`, et. `FROM` Pour plus d'informations sur l'optimisation de l'insertion rapide, consultez la section Améliorations des performances d'Amazon Aurora MySQL.

Améliorations générales :

- Correction d'un problème susceptible d'entraîner le redémarrage du serveur de base de données lorsque toutes les conditions suivantes sont définies sur true :
  - ALLOW_INVALID_DATES est désactivé en MODE SQL.
  - Le serveur de base de données traite une instruction INSERT, UPDATE, DELETE ou SELECT avec une valeur de type DATETIME non valide (le mois n'est pas compris entre 1 et 12, par exemple).
- Correction d'un problème de non-respect de la durée de conservation des journaux binaires lorsque log-bin est défini sur OFF, ce qui entraîne une utilisation du stockage plus importante que prévu. Après l'application de ce correctif, les journaux binaires seront purgés en fonction de votre durée de conservation. Pour plus d'informations sur la configuration de la durée de conservation de vos journaux binaires, consultez le Guide de l'utilisateur Aurora MySQL.
- Correction d'un problème pouvant entraîner une réduction de la mémoire disponible sur l'instance de base de données lorsque certaines instructions SQL Data Control Language (DCL) telles que GRANT ou FLUSH PRIVILEGES sont exécutées sur cette instance. L'utilisation fréquente de telles instructions peut entraîner une réduction continue de la mémoire disponible et entraîner le redémarrage de l'instance de base de données en raison de out-of-memory problèmes. L'utilisation

de ces types d'instructions sur l'instance d'enregistreur peut également entraîner une réduction de la mémoire disponible sur les instances d'enregistreur.

- Ajout d'une plus grande taille de tampon de lecture pour les lectures effectuées à partir des journaux de relais afin de minimiser le nombre d' I/O opérations de lecture, ce qui réduit les conflits entre les threads SQL I/O et SQL.
- Correction d'un problème susceptible d'entraîner l'échec de la procédure stockée mysql.rds_rotate_slow_log avec le message d'erreur « La table 'mysql.slow_log_backup' n'existe pas ».
- Correction d'un problème causé par une invalidation excessive du cache de requêtes qui entraîne une utilisation du processeur et des latences plus élevées que prévu sur le réplica en lecture, car ce dernier doit lire les données sur le disque plutôt que dans le cache de requêtes.
- Correction d'un problème qui permettait aux utilisateurs d'exécuter les commandes INSTALL PLUGIN et UNINSTALL PLUGIN sur une instance de lecteur, ce qui pouvait entraîner le blocage de LOCK_PLUGIN, LOCK_SYSTEM_Variables_Hash, LOCK_GLOBAL_SYSTEM_VARIABLES. Désormais, ces instructions ne peuvent être exécutées que sur l'instance d'enregistreur d'un cluster de base de données.
- Correction d'un problème de latence de validation des clusters plus importante que prévu lorsque la journalisation binaire est activée. Cela concerne toutes les transactions qui génèrent des événements de journal binaire volumineux (plus de 500 Mo).
- Correction d'un problème pouvant entraîner une valeur incorrecte pour la métrique trx_active_transactions de la table INFORMATION_SCHEMA.INNODB_METRICS.
- Correction d'un problème pouvant entraîner l'arrêt de la réplication logique en raison d'une incohérence du fichier binaire lors de la restauration d'une transaction importante à un point de sauvegarde.
- Les hachages d'informations d'identification sont masqués dans le journal général et dans le journal d'audit par slow-query-log défaut en utilisant un secret de masque cohérent. Il est configurable via le paramètre aurora_mask_password_hashes_type.
- Correction d'un problème en raison duquel la durée Zero-Downtime-Restart (ZDR) n'était pas correctement indiquée dans les événements observés par le client.
- Correction d'un problème qui pouvait entraîner l'échec des appels à mysql_rds_import_binlog_ssl_material avec l'erreur de serveur MySQL 1457.
- Correction d'un problème d'interblocage de l'initialisation du thread de vidage avec le thread de purge des journaux binaires. La rotation du fichier binaire actif peut alors être interrompue et ce

dernier peut continuer à croître ou provoquer des problèmes avec les nouvelles connexions au réplica du journal binaire.

- Correction d'un problème de renvoi d'un résultat obsolète sur un réplica en lecture Aurora de la part du cache de requêtes.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 5.7 incluse. Pour plus d'informations, consultez Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

- Correction d'un problème causé par le code de lecture des informations sur le jeu de caractères issues des tables d'événements d'une instruction du schéma de performance (par exemple, events_statements_current) qui n'empêche pas l'écriture simultanée de ces informations sur le jeu de caractères. Le jeu de caractères du texte des requêtes SQL pouvait alors être invalide, ce qui pouvait entraîner un arrêt du serveur. Avec ce correctif, un jeu de caractères non valide entraîne la troncature de la colonne SQL_TEXT et empêche l'arrêt du serveur. (Bogue n° 23540008)
- InnoDB : rétroportage d'un correctif pour les bogues n° 25189192 et 84038. Correction d'un problème de mise à jour de la table du dictionnaire de données INNODB_SYS_DATAFILES par InnoDB après une opération RENAME TABLE qui déplaçait une table vers un autre schéma. Une erreur au redémarrage indiquait alors que le fichier de données du tablespace était introuvable.
- InnoDB: correction d'un problème lors duquel le serveur supprimait un index de clé étrangère défini en interne lors de l'ajout d'un nouvel index, et tentait d'utiliser un index secondaire défini sur une colonne virtuelle générée en tant qu'index de clé étrangère, provoquant ainsi l'arrêt du serveur. InnoDB permet désormais à une contrainte de clé étrangère de référencer un index secondaire défini sur une colonne générée virtuelle. (Bogue n° 23533396)
- Correction d'un problème d'interblocage engendré par l'exécution simultanée par deux sessions d'une opération INSERT... ON DUPLICATE KEY UPDATE. Lors de l'annulation partielle d'un tuple, une autre session pouvait le mettre à jour. La correction de ce bogue annule les correctifs des bogues n° 11758237, 17604730 et 20040791. (Bogue n° 25966845)
- Rétroportage d'un correctif pour le bogue n°27407480 : correction d'un problème d'octroi inapproprié des privilèges EXECUTE et ALTER ROUTINE aux créateurs de routines, même lorsque automatic_sp_privileges est activé.
- Rétroportage d'un correctif pour le bogue n°24671968 : correction d'un problème lors duquel une requête pouvait produire des résultats incorrects si la clause WHERE contenait une sous-requête

dépendante, si la table comportait un index secondaire sur les colonnes de la liste de sélection, suivi des colonnes de la sous-requête, et si `GROUP BY` ou `DISTINCT` autorisait la requête à utiliser un scan restreint d'index.

- Correction d'un problème d'interruption de la réplication si une instruction de suppression de plusieurs tables est émise sur plusieurs tables comportant des clés étrangères. (Bogue n° 80821)
- Correction d'un problème lors duquel, dans des cas particuliers, certaines erreurs d'esclaves n'étaient pas ignorées même lorsque slave_skip_errors était activé. En cas d'échec de l'ouverture et du verrouillage d'une table ou en cas d'échec des conversions de champs sur un serveur exécutant une réplication basée sur des lignes, l'erreur est considérée comme critique et l'état de slave_skip_errors est ignoré. Le correctif garantit que, lorsque slave_skip_errors est activé, toutes les erreurs signalées lors de l'application d'une transaction sont correctement traitées. (Bogue n ° 70640 et 17653275)
- Correction d'un problème de réplication d'une instruction `SET PASSWORD` à partir d'un maître MySQL 5.6 vers un esclave MySQL 5.7, ou à partir d'un maître MySQL 5.7 avec la variable système log_builtin_as_identified_by_password définie sur ON vers un esclave MySQL 5.7. Le hachage du mot de passe était lui-même haché avant d'être stocké sur l'esclave. Le problème est maintenant résolu et le hachage du mot de passe répliqué est stocké tel qu'il a été initialement transmis à l'esclave. (Bogue n° 24687073)
- Correction d'un problème lors duquel la sérialisation d'une valeur JSON consistant en un sous-document volumineux enveloppé dans plusieurs niveaux de tableaux JSON et/ou d'objets nécessitait parfois un temps d'exécution excessif. (Bogue n° 23031146)
- Les instructions qui ne peuvent pas être analysées (en raison, par exemple, d'erreurs de syntaxe) ne sont plus écrites dans le journal des requêtes lentes. (Bogue n° 33732907)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.

- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

### Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

## Mises à jour du moteur de base de données Aurora MySQL du 01/11/2021 (version 2.10.3) (obsolète)

Version : 2.10.3

Aurora MySQL 2.10.3 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.10.3. Pour les clusters Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 ou version ultérieure directement vers la version 2.10.3. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge par Aurora MySQL 2.10.3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

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

Améliorations générales :

- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage du serveur de base de données en raison d'un long délai d'attente du sémaphore lorsque le thread du détecteur d'interblocages est bloqué.
- Correction d'un problème pouvant entraîner une réduction de la mémoire disponible sur l'instance de base de données lorsque certaines instructions SQL Data Control Language (DCL) telles que GRANT ou FLUSH PRIVILEGES sont exécutées sur cette instance. L'utilisation fréquente de telles instructions peut entraîner une réduction continue de la mémoire disponible et entraîner le redémarrage de l'instance de base de données en raison de out-of-memory problèmes. L'utilisation de ces types d'instructions sur l'instance d'enregistreur peut également entraîner une réduction de la mémoire disponible sur les instances d'enregistreur.
- Correction d'un problème en raison duquel les requêtes portant sur la table « performance_schema.events_waits_summary_global_by_event_name » pouvaient devenir lentes lorsqu'une instance de base de données était fortement chargée lorsque l'événement d'attente « _respond_to_client » performance_schema était activé. wait/io/aurora
- Correction d'un problème susceptible, dans de rares cas, d'entraîner le blocage et le redémarrage du serveur de base de données lorsque des transactions sont partiellement annulées en raison d'une violation de contrainte sur les index secondaires.
- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage ou le basculement de l'instance d'enregistreur lorsqu'une transaction accède à une ligne supprimée par une autre transaction.
- Correction d'un problème qui, dans de rares cas, pouvait entraîner le redémarrage de la base de données en raison d'une longue attente au sémaphore lorsque les I/O threads étaient bloqués.
- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage du réplica en lecture pendant un basculement lorsque le fichier de verrouillage des Unix Sockets est utilisé.
- Correction d'un problème causé par une invalidation excessive du cache de requêtes qui entraîne une utilisation du processeur et des latences plus élevées que prévu sur le réplica en lecture, car ce dernier doit lire les données sur le disque plutôt que dans le cache de requêtes.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

Outre les corrections ci-dessous, cette version inclut tous les correctifs de bogues jusqu'à la version 5.7 incluse. Pour plus d'informations, consultez Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x.

- Correction d'un problème causé par le code de lecture des informations sur le jeu de caractères issues des tables d'événements d'une instruction du schéma de performance (par exemple, events_statements_current) qui n'empêche pas l'écriture simultanée de ces informations sur le jeu de caractères. Le jeu de caractères du texte des requêtes SQL pouvait alors être invalide, ce qui pouvait entraîner un arrêt du serveur. Avec ce correctif, un jeu de caractères non valide entraîne la troncature de la colonne SQL_TEXT et empêche l'arrêt du serveur. (Bogue n° 23540008)
- Correction d'un problème d'arrêt du serveur lorsqu'une opération UPDATE nécessitait une table temporaire doté d'une clé primaire de plus de 1 024 octets et que cette table était créée à l'aide d'InnoDB. (Bogue n° 25153670)
- Correction d'un problème d'interblocage engendré par l'exécution simultanée par deux sessions d'une opération INSERT... ON DUPLICATE KEY UPDATE. Lors de l'annulation partielle d'un tuple, une autre session pouvait le mettre à jour. La correction de ce bogue annule les correctifs des bogues n° 11758237, 17604730 et 20040791. (Bogue n° 25966845)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 21/01/2022 (version 2.10.2) (obsolète)

Version : 2.10.2

Aurora MySQL 2.10.2 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* existant vers Aurora MySQL 2.10.0. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 existant ou d'une version supérieure directement vers la version 2.10.0.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.10.0.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- 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

Améliorations générales :

- Ajout d'une optimisation des performances pour aider à réduire la latence I/O de la base de données dans les classes d'instances 24XL.

- Ajout de la prise en charge des chiffrements SSL ECDHE. Pour plus d'informations sur la configuration de vos clients pour utiliser ces chiffrements SSL, veuillez consulter la documentation MySQL suivante : encrypted connection protocols ciphers
- Correction de problèmes de sécurité liés à l'intégration d'Aurora MySQL à d'autres AWS services tels qu'Amazon S3, Amazon ML et AWS Lambda.
- Correction d'un problème susceptible d'entraîner l'échec du redémarrage d'une instance de base de données lorsque la base de données comporte environ plus d'1 Go de combinaisons d'utilisateurs et de privilèges.
- Correction d'un problème lié à une requête parallèle susceptible d'entraîner le renvoi par la base de données de groupements ou d'ordres de tri incorrects lors de l'exécution de requêtes avec une clause GROUP BY et une clause WHERE contenant un prédicat de plage.
- Correction d'un problème entraînant l'inaccessibilité des tables general_log et slow_log après une mise à niveau de la version majeure sur place d'Aurora MySQL 1.x (compatible avec MySQL 5.6) vers Aurora MySQL 2.x (compatible avec MySQL 5.7).
- Correction d'un problème entraînant, dans de rares cas, le redémarrage de l'instance de base de données lorsque les tables innodb_trx, innodb_locks ou innodb_lockwaits sont interrogées alors que la base de données est soumise à une charge de travail importante. Les outils de surveillance tels que Performance Insights peuvent interroger ces tables.
- Correction d'un problème où la valeur d'une colonne TIMESTAMP d'une ligne existante est mise à jour avec l'horodatage le plus récent lorsque toutes les conditions suivantes sont remplies :
  1. Il existe un déclencheur pour la table.
  2. Une instruction INSERT est exécuté sur la table qui comporte une clause ON DUPLICATE KEY UPDATE.
  3. La ligne insérée provoque une violation de valeur en double dans un index UNIQUE ou CLÉ PRIMAIRE.
  4. Une ou plusieurs colonnes sont de type TIMESTAMP et affichent une valeur par défaut de CURRENT_TIMESTAMP.
- Correction d'un problème qui, dans de rares cas, pouvait empêcher un réplica de journal binaire de se connecter à une instance dont le journal binaire était activé.
- Correction d'un problème où, dans de rares conditions, les transactions n'étaient pas en mesure d'être validées lors de l'exécution sur une instance dont le journal binaire était activé.
- Correction d'un problème qui empêchait d'établir de nouvelles connexions à une instance dont le journal binaire était activé.

- Correction d'un problème qui pouvait entraîner une journalisation interne excessive lors d'une tentative de correctif sans temps d'arrêt et redémarrage sans interruption provoquant le remplissage du stockage local.
- Correction d'un problème entraînant l'arrêt d'un réplica de journal binaire avec une erreur HA_ERR_FOUND_DUPP_KEY lors de la réplication de certaines instructions DDL et DCL. Le problème se produit lorsque l'instance source est configurée avec un format de journalisation binaire MIXED et un niveau d'isolation READ COMMITTED ou READ UNCOMMITTED.
- Correction d'un problème qui empêchait le I/O thread de réplication binlog de suivre le rythme de l'instance principale lorsque la réplication multithread était activée
- Correction d'un problème à cause duquel, dans de rares cas, un nombre élevé de connexions actives à l'instance de base de données pouvait entraîner le CloudWatch CommitLatency signalement incorrect de la métrique.
- Correction d'un problème qui entraînait le remplissage du stockage local sur les instances Graviton lors de l'exécution de LOAD FROM S3 ou SELECT INTO S3.
- Correction d'un problème qui pouvait entraîner des résultats de requête erronés lors de l'interrogation d'une table avec une clé étrangère si les deux conditions suivantes étaient réunies :
  1. Cache de requête activé
  2. Une transaction comportant une suppression ou une mise à jour en cascade sur cette table est annulée
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage des instances de lecteur Aurora. Le risque que ce problème se produise augmente à mesure que le nombre d'annulations de transactions augmente.
- Correction d'un problème où le nombre d'occurrences du mutex « LOCK_epoch_id_master » dans le schéma de performances augmente lorsqu'une session est ouverte et fermée.
- Correction d'un problème susceptible d'entraîner un nombre croissant de blocages pour les charges de travail comportant de nombreuses transactions mettant à jour simultanément le même jeu de lignes.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage des instances lorsque le volume de la base de données augmente jusqu'à un multiple de 160 Go.
- Correction d'un problème lié à la requête parallèle susceptible d'entraîner le redémarrage de la base de données lors de l'exécution d'instructions SQL avec une clause LIMIT.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de l'instance de base de données lors de l'utilisation de transactions XA avec le niveau d'isolation READ COMMITTED.

- Correction d'un problème où, après le redémarrage d'une instance Aurora en lecture, celle-ci pouvait redémarrer à nouveau si elle comportait une charge de travail DDL lourde pendant le redémarrage.
- Correction d'un problème lié au signalement incorrect d'un retard de réplication de lecteur Aurora.
- Correction d'un problème qui, dans de rares conditions, peut entraîner le redémarrage d'une instance d'enregistreur en cas d'échec d'une vérification de l'intégrité des données en mémoire.
- Correction d'un problème susceptible, dans de rares conditions, d'afficher à tort le graphique « Charge de base de données » dans les sessions Performance Insights (PI) comme utilisant activement l'UC, même si les sessions ont terminé le traitement et sont inactives.
- Correction d'un problème qui, dans de rares conditions, pouvait entraîner le redémarrage du serveur de base de données lorsqu'une requête était traitée à l'aide de la requête parallèle.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de l'instance d'enregistreur dans le cluster de base de données globale en raison d'une condition de concurrence lors de la réplication de base de données globale.
- Correction d'un problème pouvait survenir lors du redémarrage d'une instance de base de données, susceptible d'entraîner plusieurs redémarrages.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Correction d'un problème dans InnoDB où une erreur de code liée aux statistiques de table déclenchait une assertion dans le fichier source dict0stats.cc. (Bogue n°24585978)
- Correction d'un problème où un index secondaire sur une colonne virtuelle devenait corrompu lorsque l'index était créé en ligne. Pour les instructions UPDATE, nous corrigeons ce problème comme suit : si la valeur de colonne virtuelle du registre d'index est définie sur NULL, nous générons cette valeur à partir du registre d'index du cluster. (Bogue n°30556595)
- Correction d'un problème dans InnoDB où la suppression de lignes marquées permettait d'acquérir un verrou de lecture externe avant la fin d'une restauration partielle. Le verrou de lecture externe empêchait la conversion d'un verrou implicite en verrou explicite pendant la restauration partielle, provoquant un échec d'assertion. (Bogue n°29195848)
- Correction d'un problème où les noms d'hôtes vides dans les comptes pouvaient entraîner un comportement erroné du serveur. (Bogue n°28653104)
- Correction d'un problème dans InnoDB où une interruption de requête pendant un temps d'attente de verrouillage provoquait une erreur. (bogue n°28068293)

- Correction d'un problème de réplication pour lequel les transactions entrelacées pouvaient parfois bloquer l'applicateur esclave lorsque le niveau d'isolement des transactions était défini sur REPEATABLE READ. (Bogue n° 25040331)
- Correction d'un problème qui pouvait entraîner le blocage des réplicas de journaux binaires en raison du délai d'attente du verrouillage. (Bogue n°27189701)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe

- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 21/10/2021 (version 2.10.1) (obsolète)

Version : 2.10.1

Aurora MySQL 2.10.1 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* existant vers Aurora MySQL 2.10.0. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 existant ou d'une version supérieure directement vers la version 2.10.0. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.10.0.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- 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

Améliorations de la disponibilité :

- Ajout de la possibilité d'arrêter proprement le cluster pour les futures mises à niveau majeures.

Améliorations générales :

- Correction d'un problème qui pouvait entraîner une consommation élevée de l'UC sur les instances de lecture en raison de la journalisation excessive des messages d'information dans les fichiers journaux de diagnostic internes.
- Correction d'un problème où la valeur d'une colonne TIMESTAMP d'une ligne existante est mise à jour avec l'horodatage le plus récent lorsque toutes les conditions suivantes sont remplies :
  1. Il existe un déclencheur pour la table.
  2. Une instruction INSERT est exécuté sur la table qui comporte une clause ON DUPLICATE KEY UPDATE.
  3. La ligne insérée provoque une violation de valeur en double dans un index UNIQUE ou CLÉ PRIMAIRE.

4. Une ou plusieurs colonnes sont de type TIMESTAMP et affichent une valeur par défaut de CURRENT_TIMESTAMP.

- Correction d'un problème introduit dans la version 2.10.0 susceptible de générer un code d'erreur lors l'utilisation de la fonction json_merge. Plus particulièrement, lorsque la fonction json_merge est utilisée dans un DDL contenant des colonnes générées, elle peut renvoyer le code d'erreur 1305.
- Correction d'un problème où, dans de rares conditions, les réplicas en lecture redémarrent lorsque l'historique des mises à jour d'un objet volumineux est en cours de validation pour la vue de lecture d'une transaction sur le réplica en lecture.
- Correction d'un problème qui, dans de rares conditions, entraîne le redémarrage d'une instance d'enregistreur en cas d'échec d'une vérification de l'intégrité des données en mémoire.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER. (Bogue n° 25209512)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 25/05/2021 (version 2.10.0) (obsolète)

Version : 2.10.0

Aurora MySQL 2.10.0 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* existant vers Aurora MySQL 2.10.0. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 existant ou d'une version supérieure directement vers la version 2.10.0.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.10.0.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- 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

Nouvelles fonctions :

- La classe d'instances `db.t3.large` est désormais prise en charge pour Aurora MySQL.
- Réplication du journal binaire :
  - Le I/O cache binlog a été introduit pour améliorer les performances des journaux binaires en réduisant les conflits entre les threads d'écriture et les threads de vidage. Pour plus d'informations, consultez Réplication avec Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
  - Dans Aurora MySQL version 2.08, nous avons introduit l'amélioration du traitement du journal binaire (binlog) pour réduire le temps de récupération sur incident et la latence de temps de validation lorsque de très grandes transactions sont incluses. Ces améliorations sont désormais prises en charge pour les clusters sur lesquels le GTID est activé.
- Plus grande disponibilité des instances de lecteur :
  - Auparavant, lors du redémarrage d'une instance d'enregistreur, toutes les instances de lecteur d'un cluster Aurora MySQL ont également redémarré. Grâce au lancement du jour, les instances de lecteur dans la région continueront de traiter des demandes de lecture lors du redémarrage d'une instance d'enregistreur, ce qui améliore la disponibilité de lecture dans le cluster. Pour plus d'informations, consultez Redémarrage d'un cluster Aurora avec disponibilité en lecture (versions 2.10 et ultérieures) dans le Guide de l'utilisateur Amazon Aurora.

    > **Important**
    >
    > Après la mise à niveau vers Aurora MySQL 2.10, le redémarrage de l'instance d'enregistreur n'effectue pas de redémarrage de l'ensemble du cluster. Si vous souhaitez redémarrer l'ensemble du cluster, vous réinitialisez maintenant toutes les instances de lecteur dans le cluster après avoir redémarré l'instance d'enregistreur.

- Amélioration des performances de lecture de pages anticipée demandées par la technique de lecture anticipée logique (LRA). Cela a été fait en regroupant les lectures de plusieurs pages dans une seule requête envoyée au stockage Aurora. Par conséquent, les requêtes utilisant l'optimisation LRA s'exécutent jusqu'à trois fois plus vite.
- Redémarrages sans interruption et application de correctifs sans temps d'arrêt :
  - Redémarrage sans interruption (ZDR) amélioré et application de correctifs sans temps d'arrêt (ZDP) pour activer ZDR et ZDP dans un plus large éventail de scénarios, y compris la prise en charge supplémentaire des cas où la journalisation binaire est activée. Aussi, une visibilité améliorée sur les événements de ZDR et de ZDP. Pour plus d'informations, consultez

Redémarrage sans interruption (ZDR) pour Amazon Aurora MySQL et Utilisation des correctifs sans temps d'arrêt dans le Guide de l'utilisateur Amazon Aurora.

Améliorations de la disponibilité :

- Améliorations pour un démarrage plus rapide lorsque la base de données comporte un grand nombre d'index temporaires et de tables créés lors d'une activité DDL interrompue antérieurement.
- Correction de plusieurs problèmes liés aux redémarrages répétés lors de la récupération sur incident des opérations DDL interrompues, telles que `DROP TRIGGER`, `ALTER TABLE` et en particulier `ALTER TABLE` qui modifie le type de partitionnement ou le nombre de partitions dans une table.
- Correction d'un problème qui pouvait entraîner le redémarrage du serveur pendant le traitement du journal DAS (Database Activity Streams).
- Correction d'un problème d'impression d'un message d'erreur lors du traitement d'une requête `ALTER` sur les tables système.

Améliorations générales :

- Correction d'un problème dans lequel le cache de requête pouvait renvoyer des résultats obsolètes sur une instance de lecteur.
- Correction d'un problème où certaines métriques de validation Aurora n'étaient pas mises à jour lorsque la variable système `innodb_flush_log_at_trx_commit` était définie sur 0 ou 2.
- Correction d'un problème où un résultat de requête stocké dans le cache de requêtes n'était pas actualisé par des transactions à plusieurs instructions.
- Correction d'un problème qui pouvait entraîner un défaut de mise à jour correcte de l'horodatage modifié pour la dernière fois des fichiers du journal binaire Cela pouvait entraîner la purge prématurée des fichiers du journal binaire, avant d'atteindre la période de rétention configurée par le client.
- Correction du nom et de la position du fichier de journal binaire signalés incorrects depuis InnoDB après une récupération sur incident.
- Correction d'un problème qui pouvait entraîner des transactions volumineuses à générer des événements de journal binaire incorrects si le paramètre `binlog_checksum` était défini sur `NONE`.
- Correction d'un problème qui provoquait l'arrêt d'un réplica de journal binaire avec une erreur si la transaction répliquée contenait une instruction DDL et un grand nombre de modifications de ligne.

- Correction d'un problème entraînant un redémarrage dans une instance de lecteur lors de la suppression d'une table.
- Correction d'un problème qui entraînait la défaillance des connecteurs open source lors de la tentative d'utilisation d'un fichier journal binaire avec une transaction volumineuse.
- Correction d'un problème qui pouvait entraîner des résultats de requête incorrects sur la colonne de géométrie volumineuse après avoir créé un index spatial sur la table avec les valeurs de géométrie volumineuse.
- La base de données recrée désormais l'espace de table temporaire pendant le redémarrage, ce qui permet de libérer et de récupérer l'espace de stockage associé.
- Correction d'un problème qui empêchait les tables `performance_schema` d'être tronquées sur les instances de lecteur Aurora.
- Correction d'un problème qui provoquait l'arrêt d'un réplica de journal binaire avec une erreur `HA_ERR_KEY_NOT_FOUND`.
- Résolution d'un problème qui entraînait le redémarrage de la base de données lors de l'exécution de l'instruction `FLUSH TABLES WITH READ LOCK`.
- Correction d'un problème qui empêchait l'utilisation de fonctions de verrouillage au niveau de l'utilisateur sur les instances de lecteur Aurora.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Les transactions entrelacées pouvaient parfois bloquer l'applicateur de réplica lorsque le niveau d'isolement des transactions était défini sur REPEATABLE READ. (Bogue n° 25040331)
- Lorsqu'une procédure stockée contenait une instruction faisant référence à une vue qui, à son tour, faisait référence à une autre vue, la procédure n'a pas pu être invoquée avec succès plus d'une fois. (Bogue n° 87858 et n° 26864199)
- Pour les requêtes comportant de nombreuses conditions `OR`, l'optimiseur est désormais plus efficace en mémoire et moins susceptible de dépasser la limite de mémoire imposée par la variable système range_optimizer_max_mem_size. En outre, la valeur par défaut de cette variable est passée de 1 536 000 à 8 388 608. (Bogue n° 79450 et n° 22283790)
- Réplication : dans la fonction `next_event()`, appelée par le thread SQL d'un réplica pour lire l'événement suivant à partir du journal de relais, le thread SQL ne libérait pas le `relaylog.log_lock` acquis lorsqu'il rencontrait une erreur (par exemple, en raison d'un journal de relais fermé). Par conséquent, tous les autres threads devaient attendre d'obtenir un verrou sur

le journal du relais à raccrocher. Avec ce correctif, le verrou est libéré avant que le thread SQL ne quitte la fonction concernée. (Bogue n° 21697821)

- Correction d'une corruption de mémoire pour `ALTER TABLE` avec une colonne virtuelle. (Bogue n° 24961167 et n° 24960450)
- Réplication : les réplicas multithreads ne pouvaient pas être configurés avec de petites files d'attente à l'aide de slave_pending_jobs_size_max s'ils avaient besoin de traiter des transactions supérieures à cette taille. Tout paquet plus grand que slave_pending_jobs_size_max était rejeté avec l'erreur `ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX`, même si le paquet était inférieur à la limite fixée par slave_max_allowed_packet. Avec ce correctif, slave_pending_jobs_size_max devient une limite flexible plutôt qu'une limite stricte. Si la taille d'un paquet dépasse slave_pending_jobs_size_max, mais est inférieure à slave_max_allowed_packet, la transaction est conservée jusqu'à ce que tous les workers de réplica aient des files d'attente vides, puis traitées. Toutes les transactions suivantes sont conservées jusqu'à ce que la transaction volumineuse soit terminée. La taille de la file d'attente des workers de réplica peut donc être limitée tout en autorisant des transactions occasionnelles plus importantes. (Bogue n° 21280753 et n° 77406)
- Réplication : lors de l'utilisation d'un réplica multithread, les erreurs d'application affichaient des données d'ID de worker incohérentes avec les données externalisées dans les tables de réplique Schéma de performances. (Bogue n° 25231367)
- Réplication : Sur une réplique de réplication basée sur le GTID exécutée avec -GTID-Mode=ON, -LOG-bin=OFF et utilisant -, lorsqu'une erreur rencontrée, qui devait être ignorée, n'était pas correctement mise à jourslave-skip-errors, ce qui entraînait une perte de synchronisation avec. `Exec_Master_Log_Pos` `Exec_Master_Log_Pos` `Read_master_log_pos` Si un `GTID_NEXT` n'était pas spécifié, le réplica ne mettrait jamais à jour son état de GTID lors du retour à partir d'une transaction d'instruction unique. Le `Exec_Master_Log_Pos` ne serait pas mis à jour, car même si la transaction était terminée, son état GTID indiquerait le contraire. Le correctif supprime la restriction de la mise à jour de l'état GTID lorsqu'une transaction est annulée uniquement si `GTID_NEXT` est spécifié. (Bogue n° 22268777)
- Réplication : une instruction partiellement échouée ne consommait pas correctement un GTID généré automatiquement ou spécifié lorsque la journalisation binaire était désactivée. Ce correctif garantit qu'un DROP TABLE, un DROP USER ou un DROP VIEW ayant partiellement échoué consomment respectivement le GTID correspondant et l'enregistrent dans les tables `@@GLOBAL.GTID_EXECUTED` et `mysql.gtid_executed` lorsque la journalisation binaire est désactivée. (Bogue n° 21686749)

- Réplication : les réplicas exécutant MySQL 5.7 ne pouvaient pas se connecter à une source MySQL 5.5 en raison d'une erreur lors de la récupération de server_uuid, qui ne faisait pas partie de MySQL 5.5. Cela était causé par des changements dans la méthode de récupération du `server_uuid`. (Bogue n° 22748612)
- Réplication : le mécanisme d'omission de transaction GTID, qui ignore silencieusement une transaction GTID précédemment exécutée, ne fonctionnait pas correctement pour les transactions XA. (Bogue n° 25041920)
- Les instructions ">XA ROLLBACK qui avaient échoué en raison d'un ID de transaction incorrect pouvaient être enregistrées dans le journal binaire avec l'ID de transaction correct et pouvaient donc être traitées par des réplicas de réplication. Une vérification est maintenant effectuée pour la situation d'erreur avant que la journalisation binaire n'ait lieu et les instructions `ROLLBACK` XA échouées ne sont pas consignées. (Bogue n° 26618925)
- Réplication : si une réplique a été configurée à l'aide d'une instruction CHANGE MASTER TO qui ne spécifiait ni le nom du fichier journal source ni la position du journal source, puis qu'elle était arrêtée avant que START SLAVE ne soit émise, puis redémarrée avec l'option - relay-log-recovery set, la réplication n'a pas démarré. Cela se produisait parce que le thread du récepteur n'avait pas été démarré avant la tentative de récupération du journal de relais. Par conséquent, aucun événement de rotation du journal n'était disponible dans le journal de relais pour fournir le nom du fichier journal source et la position du journal source. Dans ce cas, le réplica ignore désormais la récupération du journal de relais et consigne un avertissement, puis démarre la réplication. (Bogue n° 28996606 et n° 93397)
- Réplication : dans la réplication basée sur les lignes, un message qui affichait mal les longueurs de champ était renvoyé lors de la réplication à partir d'une table comportant une colonne `utf8mb3` vers une table de la même définition que celle où la colonne était définie avec un jeu de caractères `utf8mb4`. (Bogue n° 25135304 et n° 83918)
- Réplication : lorsqu'une instruction RESET SLAVE était émise sur une réplique de réplication GTIDs en cours d'utilisation, les fichiers journaux de relais existants étaient purgés, mais le nouveau fichier journal de relais de remplacement était généré avant que l'ensemble des fichiers reçus GTIDs pour le canal n'ait été effacé. L'ancien ensemble GTID a donc été écrit dans le nouveau fichier journal du relais en tant qu'`PREVIOUS_GTIDS`événement, provoquant une erreur fatale lors de la réplication indiquant que la réplique contenait GTIDs plus que la source, même si l'ensemble gtid_executed pour les deux serveurs était vide. Désormais, lorsqu'il `RESET SLAVE` est émis, l'ensemble des données reçues GTIDs est effacé avant que le nouveau fichier journal de relais ne soit généré, de sorte que cette situation ne se produise pas. (Bogue n° 27411175)

- Réplication : GTIDs lors de l'utilisation pour la réplication, les transactions comprenant des instructions à l'origine d'une erreur d'analyse (ER_PARSE_ERROR) ne pouvaient pas être ignorées manuellement par la méthode recommandée consistant à injecter une transaction vide ou de remplacement avec le même GTID. Cette action doit permettre au réplica d'identifier le GTID comme déjà utilisé et donc d'ignorer la transaction indésirable qui a partagé son GTID. Toutefois, dans le cas d'une erreur d'analyse, parce que l'instruction a été analysée avant que le GTID ne soit vérifié pour voir s'il devait être ignoré, le thread qui a appliqué la réplication s'est arrêté en raison de l'erreur d'analyse, même si l'intention était d'ignorer la transaction malgré tout. Avec ce correctif, le thread qui a appliqué la réplication ignore désormais les erreurs d'analyse si la transaction concernée doit être ignorée, car le GTID était déjà utilisé. Notez que ce changement de comportement ne s'applique pas dans le cas d'applications constituées d'une sortie de journal binaire produite par `mysqlbinlog`. Dans ce cas, il y aurait un risque qu'une transaction comportant une erreur d'analyse qui suit immédiatement une transaction ignorée soit également ignorée, alors qu'elle devrait générer une erreur. (Bogue n° 27638268)
- Réplication : activez le thread SQL pour que GTID ignore une transaction partielle. (Bogue n ° 25800025)
- Réplication : lorsqu'un paramètre de délai d'expiration négatif ou fractionnaire avait été fourni à `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()`, le serveur se comportait de manière inattendue. Avec ce correctif :
  - une valeur de délai d'expiration fractionnaire est lue telle quelle, sans arrondi ;
  - une valeur de délai d'expiration négatif est rejetée avec une erreur si le serveur est en mode SQL strict. Si le serveur n'est pas en mode SQL strict, la valeur fait en sorte que la fonction renvoie `NULL` immédiatement et sans attendre, puis émette un avertissement. (Bogue n ° 24976304 et n° 83537)
- Réplication : si la fonction `WAIT_FOR_EXECUTED_GTID_SET()` avait été utilisée avec une valeur de délai d'expiration incluant une partie fractionnée (par exemple, 1,5), une erreur dans la logique de conversion de types signifiait que le délai d'expiration était arrondi à la seconde entière la plus proche et à zéro pour les valeurs inférieures à 1 seconde (par exemple, 0,1). La logique de conversion de types a maintenant été corrigée de sorte que la valeur du délai d'expiration soit appliquée comme indiqué initialement sans arrondi. Merci à Dirkjan Bussink pour sa contribution. (Bogue n° 29324564 et n° 94247)
- GTIDs Lorsque cette option est activée, XA COMMIT sur une transaction XA déconnectée dans le cadre d'une transaction à instructions multiples a généré une assertion. (Bogue n° 22173903)
- Réplication : une assertion était déclenchée dans les builds de débogage si une instruction XA ROLLBACK était émise pour un identifiant de transaction inconnu lorsque la valeur gtid_next avait

été définie manuellement. Le serveur ne tente plus de mettre à jour l'état du GTID si une instruction `ROLLBACK XA` échoue avec une erreur. (Bogue n° 27928837 et n° 90640)

- Correction d'un problème d'ordre de tri erroné lorsque plusieurs fonctions `CASE` sont utilisées dans la clause `ORDER BY` (Bogue n° 22810883).
- Certaines requêtes utilisant le tri pouvaient accéder à une colonne non initialisée lors de l'optimisation et provoquer l'arrêt du serveur. (Bogue n° 27389294)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 12/11/2022 (version 2.09.3) (obsolète)

Version : 2.09.3

Aurora MySQL 2.09.3 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* existant vers Aurora MySQL 2.10.0. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 existant ou d'une version supérieure directement vers la version 2.10.0. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.10.0.

Pour créer un cluster avec une version plus ancienne d'Aurora MySQL, spécifiez la version du moteur via l' AWS Management Console, l' AWS CLI ou l'API Amazon RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- 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

Améliorations de la disponibilité :

- Introduction d'une optimisation susceptible de réduire la contention pour les requêtes exécutées sur des tables dans information_schema.
- Ajout de la prise en charge des chiffrements SSL ECDHE.

Améliorations générales :

- Correction d'un problème qui, dans de rares conditions, peut entraîner le redémarrage d'une instance d'enregistreur en cas d'échec d'une vérification de l'intégrité des données en mémoire.
- Correction d'un problème qui, dans de rares conditions, peut entraîner le redémarrage de l'instance de base de données lorsque le volume du cluster se développe alors que la journalisation binaire est activée.
- Correction d'une rare condition de concurrence lors du redémarrage d'une instance de base de données, susceptible d'entraîner plusieurs redémarrages.
- Correction d'un problème susceptible d'entraîner l'échec du redémarrage d'une instance de base de données lorsque la base de données comporte un grand nombre de combinaisons d'utilisateurs et de privilèges.
- Correction d'un problème lié à la requête parallèle susceptible d'entraîner le redémarrage de la base de données lors de l'exécution d'instructions SQL avec la clause LIMIT.
- Correction d'un problème lié au signalement incorrect d'un retard de réplication Aurora.
- Correction d'un problème susceptible d'entraîner l'inaccessibilité des tables general_log et slow_log après la mise à niveau de la version majeure sur place d'Aurora-MySQL 1.x (basée sur MySQL 5.6) vers Aurora-MySQL 2.x (basée sur MySQL 5.7).
- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage de l'instance de base de données lorsque les tables innodb_trx, innodb_locks ou innodb_lockwaits sont interrogées alors que la base de données est soumise à une charge de travail importante. Les outils et fonctions de surveillance tels que Performance Insights peuvent interroger ces tables.
- Correction d'un problème susceptible d'entraîner le redémarrage d'une instance de base de données lors de l'exécution de l'instruction SQL « FLUSH TABLES WITH READ LOCK ».
- Correction d'un problème susceptible d'entraîner une interruption du processus de purge InnoDB lors de la suppression d'une instance de lecteur, provoquant une augmentation temporaire de la longueur de la liste de l'historique.
- Correction d'un problème lié à une requête parallèle susceptible d'entraîner le redémarrage de la base de données lors de l'exécution d'une instruction SQL sur une table contenant une colonne virtuelle.

- Correction d'un problème lié à une requête parallèle susceptible d'entraîner le renvoi de groupements ou d'ordres de tri incorrects lors de l'exécution de requêtes avec une clause GROUP BY et une clause WHERE contenant un prédicat de plage.
- Correction d'un problème lié à une requête parallèle susceptible, dans de rares conditions, d'entraîner le redémarrage de la base de données lors de l'exécution d'instructions SQL avec des fonctions JSON.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de l'instance d'enregistreur dans le cluster de base de données globale en raison d'une condition de concurrence lors de la réplication de base de données globale.
- Correction d'un problème susceptible d'entraîner l'arrêt d'un réplica de journal binaire avec une erreur HA_ERR_FOUND_DUPP_KEY lors de la réplication de certaines instructions DDL et DCL. Le problème se produit lorsque l'instance source est configurée avec un format de journalisation binaire MIXED et un niveau d'isolation READ COMMITTED ou READ UNCOMMITTED.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de l'instance de base de données lors de l'utilisation de transactions XA au niveau d'isolation READ COMMITTED.
- Correction d'un problème où la valeur d'une colonne TIMESTAMP d'une ligne existante est mise à jour avec l'horodatage le plus récent lorsque toutes les conditions suivantes sont réunies : 1. Un déclencheur existe pour la table ; 2. Une instruction INSERT est exécutée sur la table qui comporte une clause ON DUPLICATE KEY UPDATE ; 3. La ligne insérée provoque une violation de valeur en double dans un index UNIQUE ou PRIMARY KEY ; 4. Une ou plusieurs colonnes sont de type TIMESTAMP et affichent une valeur par défaut de CURRENT_TIMESTAMP.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage d'une instance de lecteur en raison d'un traitement de vérification incorrect.
- Correction d'un problème susceptible d'entraîner le redémarrage de l'instance de lecteur lorsque l'instance d'enregistreur augmente le volume de la base de données pour franchir des limites de taille de volume spécifiques.
- Correction d'un problème susceptible d'entraîner des temps de redémarrage plus longs pour les instances de base de données utilisant des volumes de cluster clonés.
- Correction d'un problème où le redémarrage d'une instance de base de données peut échouer une ou plusieurs fois après l'exécution d'une opération TRUNCATE TABLE sur l'instance d'enregistreur.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de l'instance de base de données.

- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de l'instance d'enregistreur lorsque le volume de la base de données augmente jusqu'à un multiple de 160 Go.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue n° 23533396 - Lors de l'ajout d'un nouvel index, le serveur a supprimé un index de clé étrangère défini en interne et a tenté d'utiliser un index secondaire défini sur une colonne générée virtuelle en tant qu'index de clé étrangère, provoquant ainsi l'arrêt du serveur. InnoDB permet désormais à une contrainte de clé étrangère de référencer un index secondaire défini sur une colonne générée virtuelle.
- Bogue n° 29550513 - Réplication : dans certaines circonstances, un problème de verrouillage dans la fonction WAIT_FOR_EXECUTED_GTID_SET () peut entraîner le blocage du serveur. Ce problème a été corrigé.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 26/02/2021 (version 2.09.2) (obsolète)

Version : 2.09.2

Aurora MySQL 2.09.2 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* existant vers Aurora MySQL 2.09.2. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau

un cluster Aurora MySQL 1.23 existant ou d'une version ultérieure directement vers la version 2.09.2. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.09.2.

Pour créer un cluster avec une version plus ancienne d'Aurora MySQL, spécifiez la version du moteur via l' AWS Management Console, l' AWS CLI ou l'API Amazon RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Les clusters Aurora MySQL prennent désormais en charge les instances EC2 R6g suivantes alimentées par des processeurs AWS Graviton2 basés sur ARM :`r6g.large`,,,,,`r6g.xlarge`,`r6g.2xlarge`. `r6g.4xlarge` `r6g.8xlarge` `r6g.12xlarge` `r6g.16xlarge` Pour plus d'informations, consultez Classes d'instances de base de données Aurora dans le Guide de l'utilisateur Amazon RDS.

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- 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

Améliorations de la disponibilité :

- Correction d'un problème introduit dans la version 2.09.0 qui pouvait entraîner une latence d'écriture élevée pendant la mise à l'échelle du volume de stockage du cluster.
- Correction d'un problème lié à la fonction de redimensionnement dynamique qui pouvait entraîner le redémarrage des réplicas en lecture Aurora.
- Correction d'un problème qui pouvait entraîner des temps d'arrêt plus longs pendant la mise à niveau de 1.23.* à 2.09.*.
- Correction d'un problème dans lequel un DDL ou un DML pouvait provoquer le redémarrage du moteur lors d'une demande de lecture anticipée d'une page.
- Correction d'un problème qui provoquait l'arrêt d'un réplica de journal binaire avec une erreur si la transaction répliquée contient une instruction DDL et un grand nombre de modifications de ligne.
- Correction d'un problème de redémarrage d'une base de données agissant en tant que réplica de journal binaire lors de la réplication d'un événement DDL sur la table `time_zone` mysql.
- Correction d'un problème qui pouvait entraîner des transactions volumineuses à générer des événements de journal binaire incorrects si le paramètre `binlog_checksum` était défini sur `NONE`.
- Correction d'un problème qui provoquait l'arrêt d'un réplica de journal binaire avec une erreur `HA_ERR_KEY_NOT_FOUND`.
- Amélioration de la stabilité globale.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 11/12/2020 (version 2.09.1) (obsolète)

Version : 2.09.1

Aurora MySQL 2.09.1 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* existant vers Aurora MySQL 2.09.1. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 existant ou d'une version ultérieure directement vers la version 2.09.1. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.09.1.

Pour créer un cluster avec une version plus ancienne d'Aurora MySQL, spécifiez la version du moteur via l' AWS Management Console, l' AWS CLI ou l'API RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- 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

Modifications incompatibles :

Cette version introduit une modification d'autorisation qui affecte le comportement de la commande `mysqldump`. Les utilisateurs doivent disposer du privilège `PROCESS` pour accéder à la table `INFORMATION_SCHEMA.FILES`. Pour exécuter la commande `mysqldump` sans aucune modification, accordez le privilège `PROCESS` à l'utilisateur de base de données auquel la commande `mysqldump` se connecte. Vous pouvez également exécuter la commande `mysqldump` avec l'option `--no-tablespaces`. Avec cette option, la sortie `mysqldump` n'inclut aucune instruction `CREATE LOGFILE GROUP` ou `CREATE TABLESPACE`. Dans ce cas, la commande `mysqldump` n'accède pas à la table `INFORMATION_SCHEMA.FILES` et vous n'avez pas besoin d'accorder l'autorisation `PROCESS`.

Améliorations de la disponibilité :

- Correction d'un problème pouvant entraîner le blocage d'une séance client lorsque le moteur de base de données rencontrait une erreur lors de la lecture ou de l'écriture sur le réseau.
- Correction d'une fuite de mémoire dans la fonction de redimensionnement dynamique, introduite dans la version 2.09.0.

Bases de données globales :

- Correction de plusieurs problèmes entraînant le redémarrage des réplicas d'une région secondaire de base de données globale lors de la mise à niveau vers la version 2.09.0 alors que le scripteur de la région principale était sur une version antérieure.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Réplication : les transactions entrelacées pouvaient parfois bloquer l'applicateur esclave lorsque le niveau d'isolement des transactions était défini sur REPEATABLE READ. (Bogue n° 25040331)
- Pour les tables ayant une colonne TIMESTAMP ou DATETIME avec une valeur CURRENT_TIMESTAMP par défaut, la colonne pouvait être initialisée à `0000-00-00 00:00:00` si la table avait un déclencheur `BEFORE INSERT`. (Bogue n° 25209512, Bogue n° 84077)
- Pour les instructions INSERT pour lesquelles la liste `VALUES` produisait des valeurs pour la deuxième ligne ou une ligne suivante à l'aide d'une sous-requête contenant une jointure, le serveur pouvait s'arrêter s'il n'avait pas réussi à résoudre les privilèges requis. (Bogue n° 23762382)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 17/09/2020 (version 2.09.0) (obsolète)

Version : 2.09.0

Aurora MySQL 2.09.0 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané d'Aurora MySQL 1.23.* vers Aurora MySQL 2.09.0. Vous avez également la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.09.0. Vous ne pouvez pas mettre à niveau directement un cluster Aurora

MySQL 1.23.* vers 2.09.0. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.09.0.

> **Important**
>
> Les améliorations apportées au stockage Aurora dans cette version limitent les chemins de mise à niveau disponibles d'Aurora MySQL 1.* à Aurora MySQL 2.09. Lorsque vous mettez à niveau un cluster Aurora MySQL 1.* vers 2.09, vous devez effectuer une mise à niveau à partir d'Aurora MySQL 1.23.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Avec cette version, vous pouvez créer des instances de base de données Amazon Aurora MySQL avec jusqu'à 128 tébioctets (TiO) de stockage. La nouvelle limite de stockage est une augmentation par rapport à la limite précédente de 64 Tio. La taille de stockage de 128 Tio prend en charge des bases de données de taille plus importante. Cette capacité n'est pas prise en charge sur les petites tailles d'instance (db.t2 ou db.t3). Un espace disque logique unique ne peut pas dépasser 64 Tio en raison des limitations InnoDB avec une taille de page de 16 Ko.

  Aurora vous avertit lorsque la taille du volume du cluster est proche de 128 Tio, afin que vous puissiez prendre des mesures avant d'atteindre la limite de taille. Les alertes apparaissent dans le journal mysql et les événements RDS dans AWS Management Console.

- Vous pouvez désormais activer ou désactiver la requête parallèle pour un cluster existant en modifiant la valeur du paramètre de cluster de base de données `aurora_parallel_query`. Vous n'avez pas besoin d'utiliser `parallelquery` pour le paramètre `--engine-mode` lors de la création du cluster.

  La requête parallèle est désormais étendue et disponible dans toutes les régions où Aurora MySQL est disponible.

  Un certain nombre d'autres améliorations de fonction et de modifications de procédure ont été apportées pour la mise à niveau et l'activation des requêtes parallèles dans un cluster Aurora. Pour plus d'informations, consultez Utilisation des requêtes parallèles pour Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Aurora redimensionne dynamiquement l'espace de stockage de votre cluster. Avec le redimensionnement dynamique, l'espace de stockage de votre cluster de base de données Aurora diminue automatiquement lorsque vous supprimez des données du cluster de base de données. Pour plus d'informations, consultez Dimensionnement du stockage dans le Guide de l'utilisateur Amazon Aurora.

  > **Note**
  >
  > La fonctionnalité de redimensionnement dynamique est déployée par étapes dans les AWS régions où Aurora est disponible. Selon la région où se trouve votre cluster, il se peut que cette fonction ne soit pas encore disponible. Pour plus d'informations, veuillez consulter l'annonce des nouveautés.

Correctifs à priorité élevée :

- Rétroportage de bogue de communauté #27659490 : SELECT AVEC UTILISATION DE PLAGE DYNAMIQUE ET DE FUSION D'INDEX NÉCESSITE TROP DE MÉMOIRE (MÉMOIRE INSUFFISANTE)
- Bogue #26881508 : MYSQL #1 : DISABLE_ABORT_ON_ERROR DANS AUTH_COMMON.H
- Rétroportage de bogue de communauté #24437124 : POSSIBLE DÉPASSEMENT DE MÉMOIRE TAMPON LORS D'UNE CRÉATION DE TABLE
- Rétroportage de bogue #27158030 : INCIDENT INNODB ONLINE ALTER AVEC DML SIMULTANÉ

- Bogue #29770705 : INCIDENT SERVEUR PENDANT L'EXÉCUTION DE SELECT AVEC CLAUSE WHERE SPÉCIFIQUE
- Rétroportage de bogue #26502135 : MYSQLD SEGFAULTS DANS MDL_CONTEXTE::RY_ACQUIRE_LOCK_IMPL
- Rétroportage de bogue #26935001 : AUTO_INCREMENT DE MODIFICATION DE TABLE TENTE DE LIRE L'INDEX À PARTIR D'UN ESPACE DE TABLE SUPPRIMÉ
- Bogue #28491099 : [FATAL] BLOC DE MÉMOIRE NON VALIDE | INNODB : ÉCHEC D'ASSERTION : UT0UT.CC:670
- Bogue #30499288 : GCC 9.2.1 RENVOIE UN NOUVEL AVERTISSEMENT POUR OS_FILE_GET_PARENT_DIR
- Bogue #29952565 où MYSQLD A OBTENU UN SIGNAL 11 LORS DE L'EXÉCUTION D'UNE REQUÊTE (UNION + ORDER BY + SUB-QUERY)
- Bogue #30628268 : INCIDENT DE MÉMOIRE INSUFFISANTE
- Bogue #30441969 : BOGUE #29723340 : INCIDENT MYSQL SERVER APRÈS REQUÊTE SQL AVEC ?AST DE DONNÉES
- Bogue #30569003 : RUPTURE DE RÉPLICATION 5.7 AVEC ERREUR DE SYNTAXE AVEC GESTION DES OCTROIS
- Bogue #29915479 : L'EXÉCUTION DE COM_REGISTER_SLAVE SANS COM_BINLOG_DUMP PEUT GÉNÉRER L'ARRÊT DU SERVEUR
- Bogue #30569003 : RUPTURE DE RÉPLICATION 5.7 AVEC ERREUR DE SYNTAXE AVEC GESTION DES OCTROIS
- Bogue #29915479 : L'EXÉCUTION DE COM_REGISTER_SLAVE SANS COM_BINLOG_DUMP PEUT GÉNÉRER L'ARRÊT DU SERVEUR
- Bogue #20712046 : AFFICHAGE DES TABLES PROCESSLIST ET PERFORMANCE_SCHEMA NE MASQUE PAS LE MOT DE PASSE DEPUIS LA REQUÊTE
- Bogue de rétroportage #18898433 : PERFORMANCES EXTRÊMEMENT LENTES AVEC JOINTURES EXTERNES ET TAMPON DE JOINTURE (corrigé en 5.7.21) Les requêtes avec de nombreuses jointures gauches étaient lentes si la mise en mémoire tampon de jointure était utilisée (par exemple, à l'aide de l'algorithme de boucle imbriquée par bloc). (Bogue #18898433, Bogue #72854)"
- Bogue de rétroportage #26402045 : MYSQLD SE BLOQUE SUR LA REQUÊTE (corrigé dans MySQL 5.7.23). Certains cas de matérialisation de sous-requête pouvaient provoquer l'arrêt d'un serveur. Ces requêtes produisent maintenant une erreur suggérant que la matérialisation doit être désactivée. (Bogue #26402045)

- [Rétroportage depuis MySQL] Les utilisateurs autres que rdsadmin ne sont pas autorisés à mettre à jour la table pfs dans le réplica du lecteur.
- Correction du problème dans lequel le client ne peut pas mettre à jour le perfschema dans le réplica du lecteur
- Bogue n° 26666274 : BOUCLE INFINITE DANS LE CONTAINER DE MÉMOIRE TAMPON DE SCHÉMA DE PERFORMANCE
- Bogue #26997096 : la valeur de relay_log_space n'est pas mise à jour de manière synchronisée, de sorte qu'elle est parfois beaucoup plus élevée que l'espace disque réel utilisé par les journaux relais.
- BOGUE #25082593 : LA VALIDATION DE CLÉ ÉTRANGÈRE N'A PAS BESOIN D'ACQUÉRIR UN VERROUILLAGE D'ÉCART EN LECTURE VALIDÉE
- 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

Améliorations de la disponibilité :

- Activez le correctif ABA du gestionnaire de verrous par défaut.

- Correction d'un problème lié au gestionnaire de verrous, où une condition de concurrence pouvait provoquer le partage d'un verrou par deux transactions, provoquant le redémarrage de la base de données.
- Correction d'un problème lors de la création d'une table temporaire avec format de ligne compressé pouvant entraîner un redémarrage.
- Correction de la valeur par défaut de `table_open_cache` sur les instances 16XL et 24XL qui pouvait provoquer des basculements répétés et une utilisation élevée du processeur sur les classes d'instances volumineuses (R4/R5-16XL, R5-12XL, R5-24XL). Cela a eu un impact sur 2.07.x.
- Correction d'un problème qui faisait que la restauration d'un cluster depuis Amazon S3 vers Aurora MySQL version 2.08.0 prenait plus de temps que prévu lorsque la sauvegarde S3 n'incluait pas la table `mysql.host`.
- Correction d'un problème qui pouvait provoquer des basculements répétés en raison des mises à jour de colonne virtuelle avec des index secondaires.
- Correction d'un problème lié à la gestion de la mémoire de verrouillage des transactions avec des transactions d'écriture longues entraînant un redémarrage de la base de données.
- Correction de plusieurs problèmes dans lesquels le moteur pouvait se bloquer pendant l'application de correctifs sans temps d'arrêt tout en vérifiant le point sûr pour l'application de correctifs.
- Correction d'un problème pour ignorer la journalisation redo pour les tables temporaires, qui provoquait auparavant un plantage.
- Correction d'une condition de course dans le gestionnaire de verrous entre le fait de tuer connection/query et la fin de la session.
- Correction d'un problème où la base de données pouvait se bloquer en cas de réplica binlog et recevoir un événement DDL sur la table MySQL `time_zone`.

Bases de données globales :

- La vue MySQL `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` dans une région secondaire affiche désormais les entrées pour les réplicas appartenant à cette région.
- Correction d'échecs de requête inattendus qui pouvaient se produire dans une région secondaire de base de données globale après des problèmes de connectivité réseau temporaires entre les régions principale et secondaire.
-

Requête parallèle :

- Correction du plan `EXPLAIN` d'une requête de requête parallèle, ce qui est incorrect pour une requête simple à table unique.
- Correction de l'auto-blocage qui peut se produire lorsque la requête parallèle est activée.

Améliorations générales :

- L'exportation vers S3 prend désormais en charge le mot-clé `ENCRYPTION`.
- Le paramètre `aurora_binlog_replication_max_yield_seconds` a maintenant une valeur maximale de 36 000. Précédemment, la valeur maximale acceptée était 45. Ce paramètre ne fonctionne que lorsque le paramètre `aurora_binlog_use_large_read_buffer` est défini sur 1.
- Comportement modifié pour mapper `MIXED` `binlog_format` sur `ROW` au lieu de `STATEMENT` lors de l'exécution de `LOAD DATA FROM INFILE | S3`.
- Correction d'un problème dans lequel un réplica de journal binaire connecté à un principal de journal binaire Aurora MySQL pouvait afficher des données incomplètes lorsque le principal exécutait `LOAD DATA FROM S3` et `binlog_format` défini sur `STATEMENT`.
- Augmentation de la longueur maximale autorisée pour les variables du système d'audit `server_audit_incl_users` et `server_audit_excl_users` de 1024 octets à 2000 octets.
- Correction d'un problème qui empêchait les utilisateurs d'accéder à la base de données en cas d'abaissement du paramètre `max_connections` dans le groupe de paramètres lorsque les connexions actuelles sont supérieures à la valeur définie.
- Correction d'un problème dans les flux d'activité de données où un guillemet simple et une barre oblique inverse n'étaient pas correctement échappés.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue #27659490 : SELECT AVEC UTILISATION DE PLAGE DYNAMIQUE ET DE FUSION D'INDEX NÉCESSITE TROP DE MÉMOIRE (MÉMOIRE INSUFFISANTE)
- Bogue #26881508 : MYSQL #1 : DISABLE_ABORT_ON_ERROR DANS AUTH_COMMON.H
- Bogue #24437124 : DÉPASSEMENT DE MÉMOIRE TAMPON POSSIBLE LORS DE LA CRÉATION DE TABLE
- Bogue #27158030 : INCIDENT INNODB ONLINE ALTER AVEC DML SIMULTANÉ

- Bogue #29770705 : INCIDENT SERVEUR PENDANT L'EXÉCUTION DE SELECT AVEC CLAUSE WHERE SPÉCIFIQUE
- Bogue #26502135 : MYSQLD SEGFAULT DANS MDL_CONTEXTE::TRY_ACQUIRE_LOCK_IMPL
- Bogue #26935001 : AUTO_INCREMENT DE MODIFICATION DE TABLE TENTE DE LIRE L'INDEX À PARTIR D'UN ESPACE DE TABLE SUPPRIMÉ
- Bogue #28491099 : [FATAL] BLOC DE MÉMOIRE NON VALIDE | INNODB : ÉCHEC D'ASSERTION : UT0UT.CC:670
- Bogue #30499288 : GCC 9.2.1 RENVOIE UN NOUVEL AVERTISSEMENT POUR OS_FILE_GET_PARENT_DIR
- Bogue #29952565 : où MYSQLD A OBTENU UN SIGNAL 11 LORS DE L'EXÉCUTION D'UNE REQUÊTE (UNION + ORDER BY + SUB-QUERY)
- Bogue #30628268 : INCIDENT DE MÉMOIRE INSUFFISANTE
- Bogue #30441969 : BOGUE #29723340 : INCIDENT MYSQL SERVER APRÈS REQUÊTE SQL AVEC ?AST DE DONNÉES
- Bogue #30569003 : RUPTURE DE RÉPLICATION 5.7 AVEC ERREUR DE SYNTAXE AVEC GESTION DES OCTROIS
- Bogue #29915479 : L'EXÉCUTION DE COM_REGISTER_SLAVE SANS COM_BINLOG_DUMP PEUT GÉNÉRER L'ARRÊT DU SERVEUR
- Bogue #30569003 : RUPTURE DE RÉPLICATION 5.7 AVEC ERREUR DE SYNTAXE AVEC GESTION DES OCTROIS
- Bogue #29915479 : L'EXÉCUTION DE COM_REGISTER_SLAVE SANS COM_BINLOG_DUMP PEUT GÉNÉRER L'ARRÊT DU SERVEUR
- Bogue #20712046 : AFFICHAGE DES TABLES PROCESSLIST ET PERFORMANCE_SCHEMA NE MASQUE PAS LE MOT DE PASSE DEPUIS LA REQUÊTE
- Bogue #18898433 : PERFORMANCES EXTRÊMEMENT LENTES AVEC JOINTURES EXTERNES ET TAMPON DE JOINTURE (corrigé en 5.7.21)
- Bogue #26402045 : MYSQLD SE BLOQUE SUR LA REQUÊTE (corrigé dans MySQL 5.7.23).
- Bogue n° 23103937 : PS_TRUNCATE_ALL_TABLES() NE FONCTIONNE PAS EN MODE SUPER_READ_ONLY
- Bogue n° 26666274 : BOUCLE INFINITE DANS LE CONTAINER DE MÉMOIRE TAMPON DE SCHÉMA DE PERFORMANCE
- Bogue #26997096 : la valeur de relay_log_space n'est pas mise à jour de manière synchronisée, de sorte qu'elle est parfois beaucoup plus élevée que l'espace disque

réel utilisé par les journaux relai. (https://github.com/mysql/mysql-server/commit/78f25d2809ad457e81f90342239c9bc32a36cdfa)

- Bogue #25082593 : LA VALIDATION DE CLÉ ÉTRANGÈRE N'A PAS BESOIN D'ACQUÉRIR UN VERROUILLAGE D'ÉCART EN LECTURE VALIDÉE
- Bogue #24764800 : ÉCHEC DE LA RÉPLICATION SUR L'ESCLAVE AVEC ERREUR XAER_RMFAIL.
- Bug #81441 : AVERTISSEMENT CONCERNANT LOCALHOST LORS DE L'UTILISATION SKIP-NAME-RESOLVE.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des

performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 06/01/2022 (version 2.08.4) (obsolète)

Version : 2.08.4

Aurora MySQL 2.08.4 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau un cluster de bases de données Aurora MySQL 2.* existant vers Aurora MySQL 2.10.0. Pour les clusters exécutant Aurora MySQL version 1, vous pouvez mettre à niveau un cluster Aurora MySQL 1.23 existant ou d'une version supérieure directement vers la version 2.10.0. Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.10.0.

Pour créer un cluster avec une version plus ancienne d'Aurora MySQL, spécifiez la version du moteur via l' AWS Management Console, l' AWS CLI ou l'API Amazon RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité et améliorations générales :

- Correction de problèmes de sécurité liés à l'intégration d'Aurora MySQL à d'autres AWS services tels qu'Amazon S3, Amazon ML et AWS Lambda.
- Correction d'un problème où la valeur d'une colonne TIMESTAMP d'une ligne existante est mise à jour avec l'horodatage le plus récent lorsque toutes les conditions suivantes sont réunies : 1. Un déclencheur existe pour la table ; 2. Une instruction INSERT est exécutée sur la table qui comporte une clause ON DUPLICATE KEY UPDATE ; 3. La ligne insérée provoque une violation de valeur en double dans un index UNIQUE ou PRIMARY KEY ; 4. Une ou plusieurs colonnes sont de type TIMESTAMP et affichent une valeur par défaut de CURRENT_TIMESTAMP.
- Correction d'un problème qui, dans de rares conditions, entraîne le redémarrage d'une instance d'enregistreur en cas d'échec d'une vérification de l'intégrité des données en mémoire.
- Correction d'un problème lié à la requête parallèle susceptible d'entraîner le redémarrage de la base de données lors de l'exécution d'instructions SQL avec une clause LIMIT.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 12/11/2022 (version 2.08.3) (obsolète)

Version : 2.08.3

Aurora MySQL 2.08.3 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants directement vers Aurora MySQL 2.08.3. Vous pouvez mettre à niveau un cluster Aurora MySQL 1.* existant directement vers la version 2.07.3 ou une version ultérieure, puis directement mettre à niveau vers la version 2.08.3.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- 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

Modifications incompatibles :

Cette version introduit une modification d'autorisation qui affecte le comportement de la commande `mysqldump`. Les utilisateurs doivent disposer du privilège `PROCESS` pour accéder à la table `INFORMATION_SCHEMA.FILES`. Pour exécuter la commande `mysqldump` sans aucune modification, accordez le privilège `PROCESS` à l'utilisateur de base de données auquel la commande `mysqldump` se connecte. Vous pouvez également exécuter la commande `mysqldump` avec l'option `--no-tablespaces`. Avec cette option, la sortie `mysqldump` n'inclut aucune instruction `CREATE LOGFILE GROUP` ou `CREATE TABLESPACE`. Dans ce cas, la commande `mysqldump` n'accède pas à la table `INFORMATION_SCHEMA.FILES` et vous n'avez pas besoin d'accorder l'autorisation `PROCESS`.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue n° 23762382 - INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR.
- Bogue n° 25209512 - CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication

- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 28/08/2020 (version 2.08.2) (obsolète)

Version : 2.08.2

Aurora MySQL 2.08.2 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.08.2. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.08.2. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.08.2. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.08.2. Pour plus d'informations sur la restauration d'instantanés, consultez Restauration à partir d'un instantané de cluster de base de données dans le Guide de l'utilisateur Amazon Aurora.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs critiques :

- Correction d'un problème susceptible de provoquer une panne non planifiée et d'affecter la disponibilité de la base de données.

Correctifs de disponibilité :

- Correction d'un problème de redémarrage de la base de données Aurora MySQL s'il s'agit d'un réplica binlog et réplique un événement DDL sur la table `time_zone` mysql.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 18/06/2020 (version 2.08.1) (obsolète)

Version : 2.08.1

Aurora MySQL 2.08.1 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.08.1. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.08.1. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.08.1. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.08.1.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Transfert global dans base de données d'écriture. Dans une base de données Aurora globale, vous pouvez désormais effectuer certaines opérations d'écriture, notamment des instructions DML, lorsque vous êtes connecté à un cluster secondaire. Les opérations d'écriture sont transférées vers le cluster principal et toutes les modifications sont répliquées vers les clusters secondaires. Pour plus d'informations, consultez Utilisation du transfert d'écriture dans une base de données Amazon Aurora globale dans le Guide de l'utilisateur Amazon Aurora.

Correctifs de stabilité générale :

- Correction d'un problème survenant lors de la restauration d'un cluster Amazon S3 à partir d'Aurora MySQL version 2.08.0, laquelle prenait plus de temps que prévu si la sauvegarde S3 n'incluait pas la table `mysql.host`.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 06/02/2020 (version 2.08.0) (obsolète)

Version : 2.08.0

Aurora MySQL 2.08.0 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.08.0. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.08.0. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.08.0. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.08.0.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Amélioration du traitement du journal binaire (binlog) pour réduire le temps de récupération sur incident et la latence de temps de validation lorsque de très grandes transactions sont incluses.
- Lancement de la fonctionnalité Flux d'activité de base de données (DAS) pour Aurora MySQL. Cette fonctionnalité fournit un flux de near-real-time données sur l'activité de la base de données

dans votre base de données relationnelle pour vous aider à surveiller l'activité. Pour plus d'informations, consultez Surveillance d'Amazon Aurora à l'aide des flux d'activité de base de données dans le Guide de l'utilisateur Amazon Aurora.

- Fichiers de fuseau horaire mis à jour pour prendre en charge le dernier changement de fuseau horaire au Brésil.
- De nouveaux mots clés ont été introduits dans SQL pour exercer la fonctionnalité de jointure par hachage pour une table and/or interne spécifique : `HASH_JOINHASH_JOIN_PROBING`, et`HASH_JOIN_BUILDING`. Pour plus d'informations, consultez Indicateurs Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Introduction de la prise en charge de l'indicateur d'ordre des jointures dans Aurora MySQL version 5.7 par rétroportage d'une fonctionnalité MySQL 8.0. Les nouveaux indicateurs sont `JOIN_FIXED_ORDER`, `JOIN_ORDER`, `JOIN_PREFIX` et `JOIN_SUFFIX`. Pour une documentation détaillée de la prise en charge des indicateurs d'ordre des jointures, consultez WL #9158 : indicateurs d'ordre des jointures.
- Aurora Machine Learning prend désormais en charge les fonctions définies par l'utilisateur avec `MEDIUMINT` comme type de retour.
- La procédure `lambda_async()` stockée prend désormais en charge tous les caractères `utf8` MySQL.

Correctifs à priorité élevée :

- Correction d'un problème qui pouvait amener une instance de base de données de lecteur à renvoyer des résultats incomplets pour une requête FTS après que la table `INFORMATION_SCHEMA.INNODB_SYS_TABLES` ait été interrogée sur l'instance de base de données de rédacteur.
- CVE-2019-5443
- CVE-2019-3822

Améliorations de la disponibilité :

- Correction d'un problème qui entraînait un redémarrage de la base de données après l'exécution d'une instruction à plusieurs requêtes qui accède à plusieurs tables ou bases de données avec le cache de requête activé.
- Correction d'une condition de concurrence dans le gestionnaire de verrous qui entraînait le redémarrage de la base de données ou le basculement pendant la restauration d'une transaction.

- Correction d'un problème qui entraînait le redémarrage ou le basculement de la base de données lorsque plusieurs connexions tentaient de mettre à jour la même table avec un index de recherche en texte intégral.
- Correction d'un problème susceptible de déclencher un redémarrage ou un basculement de la base de données au cours d'une commande `kill session`. Si vous rencontrez ce problème, contactez le AWS support pour activer ce correctif sur votre instance.
- Correction d'un problème qui entraînait le redémarrage de l'instance de base de données de lecteur lors d'une transaction à plusieurs instructions avec plusieurs instructions `SELECT` et une forte charge de travail d'écriture sur l'instance de base de données de rédacteur avec `AUTOCOMMIT` activé.
- Correction d'un problème qui entraînait le redémarrage de l'instance de base de données de lecteur après l'exécution de requêtes de longue durée alors que l'instance de base de données de rédacteur se trouvait sous une forte charge de travail d'écriture OLTP.

Améliorations générales :

- Amélioration du temps de récupération de base de données et de la latence de validation pour les transactions de longue durée lorsque le journal binaire est activé.
- Amélioration de l'algorithme pour générer de meilleures statistiques pour l'estimation du nombre de valeurs distinctes sur les colonnes indexées, y compris les colonnes avec des distributions de données biaisées.
- Réduction du temps de réponse et d'utilisation de l'UC des requêtes de conjointes qui accèdent aux tables temporaires MyISAM et les résultats se déversent sur le stockage local.
- Correction d'un problème qui empêchait la restauration des instantanés d'Aurora MySQL version 5.6 avec une base de données ou des espaces contenant des noms de table vers un nouveau cluster Aurora MySQL 5.7.
- Inclut les informations de transaction de la victime lorsque le blocage est résolu dans `show engine innodb status`.
- Correction d'un problème qui provoquait le blocage des connexions lorsque des clients de plusieurs versions différentes étaient connectés à la même base de données et accédaient au cache de requête.
- Correction d'une fuite de mémoire résultant de plusieurs invocations du workflow ZDP (Application de correctifs sans temps d'arrêt) ou ZDR (Redémarrage sans interruption) tout au long de la durée de vie d'une instance de base de données.

- Correction d'un message d'erreur dans les opérations ZDP (Application de correctifs sans temps d'arrêt) ou ZDR (Redémarrage sans interruption) indiquant à tort que la dernière transaction a été abandonnée si l'indicateur de validation automatique est désactivé.
- Correction d'un problème dans les opérations ZDP (application de correctifs sans temps d'arrêt) qui pouvait entraîner un message d'erreur de défaillance du serveur lors de la restauration des variables de session utilisateur dans le nouveau processus de base de données.
- Correction d'un problème dans les opérations ZDP (application de correctifs sans temps d'arrêt) qui pouvait provoquer des échecs intermittents de base de données en cas de requêtes de longue durée en cours d'exécution pendant l'application de correctifs.
- Correction d'un problème à cause duquel les requêtes incluant une fonction Aurora Machine Learning renvoyaient des messages d'erreur vides en raison d'une réponse d'erreur mal gérée par des services de Machine Learning tels qu'Amazon SageMaker AI et Amazon Comprehend.
- Correction d'un problème dans la fonctionnalité de out-of-memory surveillance qui ne respectait pas une valeur personnalisée du `table_definition_cache` paramètre.
- Le message d'erreur « L'exécution de la requête a été interrompue » est renvoyé si une requête Aurora Machine Learning est interrompue. Auparavant, le message générique « Erreur interne dans le traitement de la demande ML » était renvoyé à la place.
- Correction d'un problème qui pouvait faire qu'un travailleur de journaux binaires subisse un délai de connexion lorsque le paramètre `slave_net_timeout` était inférieur au paramètre `aurora_binlog_replication_max_yield_seconds` et que la charge de travail était faible sur le cluster maître de journaux binaires.
- Amélioration de la surveillance de la progression de la récupération des journaux binaires par l'affichage de messages d'information dans le journal des erreurs selon une fréquence d'un message par minute.
- Correction d'un problème susceptible d'empêcher le signalement des transactions actives par la requête `SHOW ENGINE INNODB STATUS`.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue #25289359 : un verrou de cache de texte intégral pris lors de la synchronisation des données n'était pas libéré si la taille du cache de texte intégral dépassait la limite de taille du cache de texte intégral.
- Bogue #29138644 : la modification manuelle de l'heure système pendant l'exécution du serveur MySQL a provoqué des retards du thread de nettoyage de page.

- Bogue #25222337 : un nom de champ de colonne virtuelle NULL dans un index virtuel a provoqué une sortie du serveur lors d'une comparaison de noms de champ qui se produit lors du remplissage de colonnes virtuelles affectées par une contrainte de clé étrangère.
- Bogue #25053286 : l'exécution d'une procédure stockée contenant une requête qui a accédé à une vue pouvait allouer de la mémoire qui n'était pas libérée jusqu'à la fin de la session.
- Bogue #25586773 : l'exécution d'une procédure stockée contenant une instruction qui a créé une table à partir du contenu de certaines instructions SELECT pouvait entraîner une fuite de mémoire.
- Bogue #28834208 : au cours de l'application de journal, après une opération OPTIMISTIE TABLE, InnoDB n'a pas rempli de colonnes virtuelles avant de vérifier les mises à jour d'index de colonne virtuelle.
- Bogue #26666274 : boucle infinie dans le conteneur de mémoire tampon de schéma de performance en raison d'un dépassement d'entier non signé 32 bits.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL 15/08/15 (version 2.07.10) (obsolète)

Version : 2.07.10

Aurora MySQL 2.07.10 est disponible. Les versions 2.07 d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.*, 3.01.*, 3.02.*, 3.03.* et 3.04.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.07.10. Vous avez également la possibilité de mettre à niveau des

clusters de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.07.10. La mise à niveau sur place est disponible pour les clusters Aurora MySQL 1.* vers Aurora MySQL 2.* (voir Mise à niveau d'Aurora MySQL 1.x vers 2.x). Elle est également disponible pour les clusters Aurora MySQL 2.* vers Aurora MySQL 3.* (voir Mise à niveau d'Aurora MySQL 2.x vers 3.x).

Immédiatement après l'exécution d'une mise à niveau sur place de la version du moteur vers Aurora MySQL 2.07.10, une mise à niveau du système d'exploitation est appliquée automatiquement à toutes les instances concernées sur les classes d'instances de base de données db.r4, db.r5, db.t2 et db.t3, si les instances exécutent une ancienne version du système d'exploitation. Dans un cluster de base de données multi-AZ, toutes les instances de lecteur appliquent d'abord la mise à niveau du système d'exploitation. Lorsque la mise à niveau du système d'exploitation sur la première instance d'enregistreur est terminée, un basculement se produit et l'instance d'enregistreur précédente est mise à niveau.

> **Note**
>
> La mise à niveau du système d'exploitation n'est pas appliquée automatiquement aux bases de données globales Aurora lors des mises à niveau majeures des versions.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

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

- CVE-2022-43551
- CVE-2022-37434
- Correction d'un problème susceptible d'empêcher l'écriture dans le journal d'audit d'événements signalés lors du traitement des rotations du journal d'audit.
- Mise à jour des chiffrements SSL par défaut utilisés par Aurora MySQL pour exclure les valeurs DES- CBC3 -SHA moins sécurisées du paramètre de base de données SSL_CIPHER. Si vous rencontrez des problèmes de connexion SSL en raison de la suppression du chiffrement DES- CBC3 -SHA, veuillez utiliser un algorithme de chiffrement sécurisé applicable parmi les informations suivantes, Configuration des suites de chiffrement pour les connexions aux clusters de bases de données Aurora MySQL.
- OpenSSL a été mis à jour vers OpenSSL version 1.0.2zh.

Améliorations générales :

- Ajout de la prise en charge des chiffrements SSL ECDHE-RSA qui utilisent des clés plus petites pour le chiffrement.
- Correction d'un problème de gestion de mémoire lors de l'exécution de requêtes avec des jointures par hachage.

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP).
- Traitement par lot des analyses. Pour de plus amples informations, veuillez consulter Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Instruction SQL `CREATE TABLESPACE`
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL 04/03/2023 (version 2.07.9) (obsolète)

Version : 2.07.9

Aurora MySQL 2.07.9 est disponible. Les versions 2.07 d'Aurora MySQL sont compatibles avec MySQL 5.7.12. Pour plus d'informations sur les modifications apportées, consultez Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 2.07.*, 2.11.*, 3.01.*, 3.02.* et 3.03.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge par Aurora MySQL 2.07.9. Vous avez également la possibilité de mettre à niveau des clusters de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.07.9. La mise à niveau sur place est disponible pour les clusters Aurora MySQL 1.* vers Aurora MySQL 2.* (voir Mise à niveau d'Aurora MySQL 1.x vers 2.x). Elle est également disponible pour les clusters Aurora MySQL 2.* vers Aurora MySQL 3.* (voir Mise à niveau d'Aurora MySQL 2.x vers 3.x).

Immédiatement après l'exécution d'une mise à niveau sur place de la version du moteur vers Aurora MySQL 2.07.9, une mise à niveau du système d'exploitation est appliquée automatiquement à toutes

les instances concernées sur les classes d'instances de bases de données db.r4, db.r5, db.t2 et db.t3, si les instances exécutent une ancienne version du système d'exploitation. Dans un cluster de base de données multi-AZ, toutes les instances de lecteur appliquent d'abord la mise à niveau du système d'exploitation. Lorsque la mise à niveau du système d'exploitation sur la première instance d'enregistreur est terminée, un basculement se produit et l'instance d'enregistreur précédente est mise à niveau.

> **Note**
>
> La mise à niveau du système d'exploitation n'est pas appliquée automatiquement aux bases de données globales Aurora lors des mises à niveau majeures des versions.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Problèmes de sécurité corrigés et CVEs répertoriés ci-dessous :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- CVE-2022-32221

Améliorations de la disponibilité :

- Correction d'un problème de réduction de la mémoire disponible à cause de la rotation du journal d'audit avancé, qui pouvait entraîner le redémarrage de l'instance de base de données.
- Correction d'un problème pouvant survenir lors du redémarrage de la base de données et qui empêchait cette dernière de démarrer correctement pendant une durée prolongée.

Améliorations générales :

- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage des instances lorsque le volume de la base de données augmente jusqu'à un multiple de 160 Go.

## Fonctions non prises en charge dans Aurora MySQL version 2

Les fonctionnalités suivantes ne sont pas prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP).
- Traitement par lot des analyses. Pour de plus amples informations, veuillez consulter Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 16/06/2022 (version 2.07.8) (obsolète)

Version : 2.07.8

Aurora MySQL 2.07.8 est disponible. Les versions 2.* d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.* d'Aurora MySQL sont compatibles avec MySQL 5.6.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.07.8. Vous avez également la possibilité de mettre à niveau des clusters de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.07.8. La mise à niveau sur place est disponible pour les clusters Aurora MySQL 1.* vers Aurora MySQL 2.* (voir Mise à niveau d'Aurora MySQL 1.x vers 2.x). Elle est également disponible pour les clusters Aurora MySQL 2.* vers Aurora MySQL 3.* (voir Mise à niveau d'Aurora MySQL 2.x vers 3.x).

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

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

Améliorations générales :

- Correction d'un problème susceptible, dans de rares cas, d'entraîner le redémarrage du serveur de base de données lorsque le thread du détecteur d'interblocages est bloqué en raison d'une condition de concurrence.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Lorsqu'une mise à jour nécessitait une table temporaire avec une clé primaire de plus de 1 024 octets et que cette table était créée à l'aide d'InnoDB, le serveur pouvait s'arrêter. (Bogue n ° 25153670)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.

- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 24/11/2021 (version 2.07.7) (obsolète)

Version : 2.07.7

Aurora MySQL 2.07.7 est disponible. Les versions 2.* d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.* d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge par Aurora MySQL 2.07.7. Vous avez également la possibilité de mettre à niveau des clusters de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.07.7. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* vers 2.07.7. En revanche, vous pouvez restaurer son instantané dans Aurora MySQL 2.07.7.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- CVE-2019-17543

- CVE-2019-2960

Améliorations générales :

- Correction de problèmes de sécurité liés à l'intégration d'Aurora MySQL à d'autres AWS services tels qu'Amazon S3, Amazon ML, Lambda.
- Correction d'un problème lié au signalement incorrect d'un retard de réplication Aurora.
- Correction d'un problème susceptible d'entraîner l'échec du redémarrage d'une instance de base de données lorsque la base de données comporte un grand nombre de combinaisons d'utilisateurs et de privilèges.
- Correction d'un problème susceptible d'entraîner l'inaccessibilité des tables general_log et slow_log après la mise à niveau de la version majeure sur place d'Aurora MySQL 1.x (basée sur MySQL 5.6) vers Aurora MySQL 2.x (basée sur MySQL 5.7).
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage d'une instance de lecteur en raison d'un traitement de vérification incorrect.
- Correction d'un problème susceptible, dans de rares conditions, d'afficher le graphique « Charge de base de données » dans les sessions Performance Insights (PI) comme utilisant activement l'UC, même si les sessions ont terminé le traitement et sont inactives.
- Correction d'un problème lié à la requête parallèle susceptible d'entraîner le redémarrage de la base de données lors de l'exécution d'instructions SQL avec une clause LIMIT.
- Correction d'un problème où la valeur d'une colonne TIMESTAMP d'une ligne existante est mise à jour avec l'horodatage le plus récent lorsque toutes les conditions suivantes sont remplies : 1. Un déclencheur existe pour la table ; 2. Une instruction INSERT est exécutée sur la table qui comporte une clause ON DUPLICATE KEY UPDATE ; 3. La ligne insérée provoque une violation de valeur en double dans un index UNIQUE ou CLÉ PRIMAIRE. ; 4. Une ou plusieurs colonnes sont de type TIMESTAMP et affichent une valeur par défaut de CURRENT_TIMESTAMP.
- Correction d'un problème susceptible, dans de rares conditions, d'entraîner le redémarrage de l'instance de base de données lors de l'utilisation de transactions XA au niveau d'isolation READ COMMITTED.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête

- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 02/09/2021 (version 2.07.6) (obsolète)

Version : 2.07.6

Aurora MySQL 2.07.6 est disponible. Les versions 2.* d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.* d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.07.6. Vous avez également la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.07.6. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* vers 2.07.6. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.07.6.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- L'INSERTION D'ENREGISTREMENTS DE TAILLE 64K PREND TROP DE TEMPS. (Bogue n° 23031146)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication

- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 06/07/2021 (version 2.07.5) (obsolète)

Version : 2.07.5

Aurora MySQL 2.07.5 est disponible. Les versions 2.* d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.* d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.07.5. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.07.5. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.07.5. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.07.5.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Améliorations de la disponibilité :

- Résolution d'un problème qui empêchait les verrous de niveau utilisateur sur un réplica Aurora.

- Résolution d'un problème qui pouvait entraîner un redémarrage d'une base de données lors de l'utilisation de transactions XA au niveau d'isolement `READ COMMITTED`.
- Longueur maximale autorisée étendue jusqu'à 2 000 pour les paramètres globaux `server_audit_incl_users` et `server_audit_excl_users`.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 04/03/2021 (version 2.07.) (obsolète)

Version : 2.07.4

Aurora MySQL 2.07.4 est disponible. Les versions 2.* d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.* d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.07.4. Vous avez également la possibilité de mettre à niveau les clusters de base de données Aurora MySQL 2.* vers Aurora MySQL 2.07.4. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.07.4. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.07.4.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

- 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

Améliorations de la disponibilité :

- Correction d'un problème qui pouvait entraîner le blocage d'un client en cas d'erreur réseau lors de la lecture ou de l'écriture d'un paquet réseau.
- Amélioration des temps de redémarrage du moteur dans certains cas après l'interruption du DDL.
- Correction d'un problème dans lequel un DDL ou un DML pouvait provoquer le redémarrage du moteur lors d'une demande de lecture anticipée d'une page.
- Correction d'un problème à cause duquel une réplique pouvait redémarrer lors de l'exécution d'un scan inversé table/index sur une réplique Aurora Read.
- Correction d'un problème dans l'opération de cluster clone qui pouvait rallonger la durée du clone.

- Correction d'un problème susceptible de provoquer le redémarrage d'une base de données lors de l'utilisation de l'optimisation des requêtes parallèles pour des colonnes géospatiales
- Correction d'un problème qui provoquait l'arrêt d'un réplica de journal binaire avec une erreur `HA_ERR_KEY_NOT_FOUND`.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Correction d'un problème dans l'analyseur n-gram de texte intégral lors de la gestion de jetons contenant « » (espace), « % » ou « , ». Les clients doivent reconstruire leurs index FTS si vous utilisez un analyseur n-gram. (Bogue n° 25873310)
- Correction d'un problème qui pouvait entraîner le redémarrage du moteur pendant l'exécution de la requête avec des vues SQL imbriquées. (Bogue n° 27214153, Bogue n° 26864199)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 10/11/2020 (version 2.07.3) (obsolète)

Version : 2.07.3

Aurora MySQL 2.07.3 est disponible. Les versions 2.* d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.* d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.07.3. Vous avez également la possibilité de mettre à niveau les

clusters de bases de données Aurora MySQL 2.* vers Aurora MySQL 2.07.3. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* vers 2.07.3. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.07.3.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré.

- 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

Modifications incompatibles :

Cette version introduit une modification d'autorisation qui affecte le comportement de la commande `mysqldump`. Les utilisateurs doivent disposer du privilège `PROCESS` pour accéder à la table `INFORMATION_SCHEMA.FILES`. Pour exécuter la commande `mysqldump` sans aucune modification, accordez le privilège `PROCESS` à l'utilisateur de base de données auquel la commande `mysqldump` se connecte. Vous pouvez également exécuter la commande `mysqldump` avec l'option `--no-tablespaces`. Avec cette option, la sortie `mysqldump` n'inclut aucune instruction `CREATE LOGFILE GROUP` ou `CREATE TABLESPACE`. Dans ce cas, la commande `mysqldump` n'accède pas à la table `INFORMATION_SCHEMA.FILES` et vous n'avez pas besoin d'accorder l'autorisation `PROCESS`.

Améliorations de la disponibilité :

- Correction d'un problème de course dans le gestionnaire de verrous entre la fin d'une session connection/query et la fin de la session, entraînant le redémarrage de la base de données.
- Correction d'un problème entraînant le redémarrage de la base de données après l'exécution d'une instruction à plusieurs requêtes qui accède à plusieurs tables ou bases de données avec le cache de requête activé.
- Correction d'un problème pouvant provoquer des redémarrages répétés en raison des mises à jour de colonne virtuelle avec des index secondaires.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- InnoDB : les transactions XA simultanées exécutées avec succès pour la phase de préparation XA sur le maître étaient en conflit lorsqu'elles étaient réutilisées sur l'esclave, entraînant ainsi un délai d'attente de verrouillage dans le thread applicateur. Le conflit était dû à la plage de verrouillage d'écart qui variait lorsque les transactions étaient réutilisées en série sur l'esclave. Pour éviter ce type de conflit, les verrous d'écart acceptés par les transactions XA au niveau d'isolement READ COMMITTED sont désormais publiés (et ne sont plus hérités) lorsque les transactions XA atteignent l'étape de préparation. (Bogue n° 27189701, Bogue n° 25866046)
- InnoDB : un verrou d'écart a été pris inutilement lors de la validation de clé étrangère et de l'utilisation du niveau d'isolement READ COMMITTED. (Bogue n° 25082593)

- Réplication : lors de l'utilisation de transactions XA, si un délai d'attente de verrouillage ou un blocage se produisait pour le thread applicateur (SQL) sur un esclave de réplication, la nouvelle tentative automatique ne fonctionnait pas, car même si le thread SQL procédait à une restauration, il ne restaurait pas la transaction XA. Par conséquent, lorsque la transaction était relancée, le premier événement était XA START et n'était pas valide étant donné que la transaction XA était déjà en cours, entraînant une erreur XAER_RMFAIL. (Bogue n° 24764800)
- Réplication : les transactions entrelacées pouvaient parfois bloquer l'applicateur esclave lorsque le niveau d'isolement des transactions était défini sur REPEATABLE READ. (Bogue n° 25040331)
- Réplication : la valeur renvoyée par une instruction SHOW SLAVE STATUS pour la taille combinée totale de tous les fichiers journaux relais existants (Relay_Log_Space) pouvait devenir beaucoup plus élevée que l'espace disque réel utilisé par les journaux relais. Le I/O thread ne verrouillait pas la variable lors de la mise à jour de la valeur, de sorte que le thread SQL pouvait automatiquement supprimer un fichier journal de relais et écrire une valeur réduite avant que le I/O thread n'ait fini de mettre à jour la valeur. Le thread d'I/O écrivait ensuite son calcul de taille d'origine, en ignorant la mise à jour du thread SQL et en rajoutant ainsi l'espace du fichier supprimé. La valeur Relay_Log_Space est désormais verrouillée pendant les mises à jour afin d'empêcher les mises à jour simultanées et de garantir un calcul précis. (Bogue n° 26997096, Bogue n° 87832)
- Pour les instructions INSERT pour lesquelles la liste VALUES produisait des valeurs pour la deuxième ligne ou une ligne suivante à l'aide d'une sous-requête contenant une jointure, le serveur pouvait s'arrêter s'il n'avait pas réussi à résoudre les privilèges requis. (Bogue n° 23762382)
- Pour les tables ayant une colonne TIMESTAMP ou DATETIME avec une valeur CURRENT_TIMESTAMP par défaut, la colonne pouvait être initialisée à `0000-00-00 00:00:00` si la table avait un déclencheur `BEFORE INSERT`. (Bogue n° 25209512, Bogue n° 84077)
- Les tentatives simultanées de plusieurs threads pour enregistrer et annuler l'enregistrement des objets de schéma de performances de métadonnées pouvaient entraîner l'arrêt d'un serveur. (Bogue n° 26502135)
- L'exécution d'une procédure stockée contenant une instruction qui avait créé une table à partir du contenu de certaines instructions SELECT pouvait entraîner une fuite de mémoire. (Bogue n° 25586773)
- L'exécution d'une procédure stockée contenant une requête qui avait accédé à une vue pouvait allouer de la mémoire qui n'était pas libérée avant la fin de la séance. (Bogue n° 25053286)
- Certains cas de matérialisation de sous-requête pouvaient provoquer l'arrêt d'un serveur. Ces requêtes produisent maintenant une erreur suggérant que la matérialisation doit être désactivée. (Bogue #26402045)

- Les requêtes avec de nombreuses jointures gauches étaient lentes si la mise en mémoire tampon de jointure était utilisée (par exemple, à l'aide de l'algorithme de boucle imbriquée par bloc). (Bogue n° 18898433, Bogue n° 72854)
- L'optimiseur ignorait la deuxième colonne d'un index composite lors de l'exécution d'une jointure interne avec une clause `LIKE` sur la deuxième colonne. (Bogue n° 28086754)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 17/04/2020 (version 2.07.2) (obsolète)

Version : 2.07.2

Aurora MySQL 2.07.2 est disponible. Les versions 2.* d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.* d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.07.2. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.07.2. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.07.2. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.07.2.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

- CVE-2016-8287
- CVE-2016-5634

Correctifs à priorité élevée :

- Correction d'un problème qui entraînait un clonage plus long sur certains clusters de bases de données avec des charges d'écriture élevées.
- Correction d'un problème qui pouvait entraîner des requêtes sur une instance de base de données de lecteur avec des plans d'exécution utilisant des index secondaires pour renvoyer des données non validées. Le problème se limitait aux données affectées par des opérations en langage de manipulation de données (DML) qui modifient les colonnes de la clé d'index primaire ou secondaire.

Améliorations générales :

- Correction d'un problème qui ralentissait la restauration d'un cluster de base de données Aurora 1.x contenant des index de recherche plein texte (FTS) vers un cluster de base de données Aurora 2.x.
- Correction d'un problème qui ralentissait la restauration d'un instantané de base de données Aurora 1.x contenant des tables partitionnées avec des caractères spéciaux dans les noms de table vers un cluster de base de données Aurora 2.x.

- Correction d'un problème qui provoquait des erreurs lors de l'interrogation des journaux de requête lentes et des journaux généraux dans les instances de base de données de lecteur.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue #23104498 : correction d'un problème dans le schéma de performance lors du signalement de la mémoire totale utilisée. (https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b)
- Bogue #22551677 : correction d'un problème dans le schéma de performance qui pouvait entraîner le plantage du moteur de base de données lorsque d'une tentative de le mettre hors ligne. (https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9)
- Bogue #23550835, Bogue #23298025, Bogue #81464 : correction d'un problème dans le schéma de performance qui provoquait un plantage du moteur de base de données en raison du dépassement de la capacité d'un tampon interne. (https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc)

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 23/12/2019 (version 2.07.1) (obsolète)

Version : 2.07.1

Aurora MySQL 2.07.1 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL actuellement prise en charge dans Aurora MySQL 2.07.1. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.07.1. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.07.1. En revanche, vous pouvez restaurer son instantané vers Aurora MySQL 2.07.1.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Chine (Ningxia) [cn-northwest-1], Asie-Pacifique (Hong Kong) [ap-east-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs à priorité élevée :

- Correction d'une fuite de mémoire lente dans le sous-système de suivi et de journalisation de base de données spécifique à Aurora qui réduisait la mémoire libérable.

Correctifs de stabilité générale :

- Correction d'un incident lors de l'exécution d'une requête complexe impliquant une agrégation et des jointures de plusieurs tables utilisant en interne des tables intermédiaires.

## Comparaison avec Aurora MySQL version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne

- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 2.07.0) (obsolète)

Version : 2.07.0

Aurora MySQL 2.07.0 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL prise en charge sur Aurora MySQL 2.07.0. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.07.0. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.07.0. En revanche, vous pouvez restaurer son instantané vers Aurora MySQL 2.07.0.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Chine (Ningxia) [cn-northwest-1], Asie-Pacifique (Hong Kong) [ap-east-1], Moyen-Orient (Bahreïn) [me-south-1] et Amérique du Sud (São Paulo) [sa-east-1] 1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Les bases de données mondiales permettent désormais d'ajouter des régions de réplication secondaires en lecture seule pour les clusters de bases de données déployés dans ces AWS régions : régions : USA Est (Virginie du Nord) [us-east-1], USA Est (Ohio) [us-east-2], USA Ouest (Californie du Nord) [us-west-1], USA Ouest (Oregon) [us-west-2], Europe (Irlande) [eu-west-2] eu-west-1], Europe (Londres) [eu-west-2], Europe (Paris) [eu-west-3], Asie-Pacifique (Tokyo) [ap-northeast-1], Asie-Pacifique (Séoul) [ap-northeast-2], Asie-Pacifique (Singapour) [ap-northeast-2] ap-southeast-1], Asie-Pacifique (Sydney) [ap-southeast-2], Canada (centre) [ca-central-1], Europe (Francfort) [ eu-central-1] et Asie-Pacifique (Mumbai) [ap-south-1].
- L'apprentissage automatique Amazon Aurora est une intégration hautement optimisée entre la base de données Aurora MySQL et les services d'apprentissage AWS automatique (ML). Le Machine Learning Aurora permet aux développeurs d'ajouter une grande variété de prédictions ML à leurs applications de bases de données en appelant les modèles ML à l'aide du langage de programmation SQL qu'ils utilisent déjà pour le développement de leurs bases de données, sans avoir à créer des intégrations personnalisées ou à découvrir des outils distincts. Pour plus d'informations, veuillez consulter Utilisation des fonctionnalités du Machine Learning (ML) avec Amazon Aurora.
- Ajout de la prise en charge pour le niveau d'isolation ANSI `READ COMMITTED` sur les réplicas en lecture. Ce niveau d'isolation permet aux requêtes de longue durée sur le réplica en lecture de s'exécuter sans impacter le haut débit des écritures sur le nœud d'écriture. Pour plus d'informations, veuillez consulter Niveaux d'isolation Aurora MySQL.

Correctifs critiques :

- CVE-2019-2922
- CVE-2019-2923

- CVE-2019-2924
- CVE-2019-2910

Correctifs à priorité élevée :

- Correction du problème de récupération de DLL qui entraînait un arrêt prolongé de la base de données. Les clusters qui deviennent indisponibles après exécution de l'instruction de suppression de plusieurs tables, par exemple `DROP TABLE t1, t2, t3`, doivent être mis à jour avec cette version.
- Correction du problème de récupération de DLL qui entraînait un arrêt prolongé de la base de données. Les clusters devenus indisponibles après l'exécution des instructions DLL `INPLACE ALTER TABLE` doivent être mis à jour avec cette version.

Correctifs de stabilité générale :

- Correction d'un problème qui générait des données non cohérentes dans la table `information_schema.replica_host_status`.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue #26251621 : COMPORTEMENT INCORRECT DU DÉCLENCHEUR ET DE GCOL
- Bug #22574695: ASSERTION `!TABLE || (!TABLE->READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL
- Bogue #25966845 : LES INSERTIONS SUR CLÉ DUPLIQUÉE GÉNÈRENT UN VERROU
- Bogue #23070734 : LES TABLES TRONQUÉES SIMULTANÉES PROVOQUENT UN ARRÊT
- Bogue #26191879 : LES CASCADES DE CLÉS ÉTRANGÈRES UTILISENT TROP DE MÉMOIRE
- Bogue #20989615 : INNODB AUTO_INCREMENT GÉNÈRE DEUX FOIS LA MÊME VALEUR

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.07.0 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.07.0 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 22/11/2019 (version 2.06.0) (obsolète)

Version : 2.06.0

Aurora MySQL 2.06.0 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Versions Aurora MySQL actuellement prises en charge : 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*, 2.05.* et 2.06.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL prise en charge sur Aurora MySQL 2.06.0. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.06.0. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.06.0. En revanche, vous pouvez restaurer son instantané vers Aurora MySQL 2.06.0.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Chine (Ningxia) [cn-northwest-1], Asie-Pacifique (Hong Kong) [ap-east-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Les clusters Aurora MySQL prennent désormais en charge les types d'instance db.r5.8xlarge, db.r5.16xlarge et db.r5.24xlarge. Pour plus d'informations sur les types d'instance pour les clusters

Aurora MySQL, consultez Classes d'instances de base de données Aurora dans le Guide de l'utilisateur Amazon Aurora.

- La fonctionnalité de jointure de hachage est désormais disponible et ne nécessite pas que le mode lab Aurora soit activé (ON). Cette fonctionnalité peut améliorer les performances de requêtes lorsque vous devez joindre une grande quantité de données au moyen d'une équijointure. Pour plus d'informations, consultez Utilisation de l'API de données pour Aurora Serverless dans le Guide de l'utilisateur Amazon Aurora.
- La fonctionnalité de conflits entre lignes critiques est désormais disponible et ne nécessite pas que le mode lab Aurora soit activé (ON). Cette fonction améliore nettement le débit pour les charges où de nombreuses transactions sont en conflit pour les lignes d'une même page.
- Aurora MySQL 2.06 et versions ultérieures prennent en charge le retour d'un cluster de base de données à une heure spécifique, sans restaurer les données à partir d'une sauvegarde. Cette fonctionnalité, connue sous le nom de Backtrack (ou retour sur suivi), offre un moyen rapide de récupérer d'erreurs d'utilisateur, comme la suppression non souhaitée d'une table ou d'une ligne. Le retour sur suivi ne dure que quelques secondes, même pour les bases de données volumineuses. Lisez le AWS blog pour obtenir une vue d'ensemble, et reportez-vous à la section Retour en arrière d'un cluster de base de données Aurora dans le guide de l'utilisateur Amazon Aurora pour plus de détails.
- Aurora 2.06 et versions ultérieures prennent en charge les appels synchrones via AWS Lambda la fonction native. `lambda_sync()` La fonction native `lambda_async()` est également disponible et peut être utilisée comme alternative à la procédure stockée existante pour l'appel Lambda asynchrone. Pour plus d'informations sur l'appel des fonctions Lambda, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Correctifs critiques :

Aucun.

Correctifs à priorité élevée :

Correctifs de sécurité

- 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

Gestion des connexions

- La disponibilité des bases de données a été améliorée afin de mieux répondre à l'augmentation du nombre de connexions clients lors de l'exécution d'une ou de plusieurs connexions DDLs. La gestion s'effectue en créant temporairement des threads supplémentaires si nécessaire. La mise à niveau est conseillée si la base de données ne répond plus suite à un pic dans les connexions, parallèlement au traitement des DDL.

Redémarrage du moteur

- Correction du problème d'indisponibilité prolongée lors du redémarrage du moteur. La correction concerne un problème d'initialisation du pool de mémoires tampon. Ce problème se produit rarement, mais peut éventuellement impacter toute version prise en charge.
- Correction d'un problème qui entraîne le redémarrage d'une base de données configurée comme journal binaire principal tandis qu'une forte charge de travail en écriture est en cours d'exécution.

Correctifs de stabilité générale :

- Les améliorations apportées à l'accès des requêtes aux données non mises en cache peuvent être plus lentes qu'à l'ordinaire. Les clients confrontés à une latence en lecture anormalement élevé alors qu'ils accèdent aux données mises en cache sont invités à procéder à une mise à niveau.

- Correction d'un problème qui entraînait la restauration de tables partitionnées à partir d'un instantané de base de données. Il est conseillé aux clients confrontés à des erreurs lors de l'accès aux tables partitionnées d'une base de données restaurée à partir d'un instantané d'une base de données Aurora MySQL 1.* d'utiliser cette version.
- Stabilité améliorée des réplicas Aurora en corrigeant la contention des verrous entre les threads traitant les requêtes en lecture et le thread appliquant les modifications du schéma tandis qu'une requête DLL est en cours sur l'instance de bases de données d'écriture.
- Correction d'un problème de stabilité lié à la mise à jour de la table `mysql.innodb_table_stats`, déclenché par les opérations DDL.
- Correction d'un problème signalant à tort une erreur `ERROR 1836` lorsqu'une requête imbriquée est exécutée sur une table temporaire sur le réplica Aurora.

Améliorations des performances :

- Amélioration des performances de la réplication des journaux binaires en empêchant les appels d'API superflus au cache si le cache des requêtes a été désactivé sur le travailleur de journaux binaires.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.06.0 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.06.0 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 11/11/2019 (version 2.05.0) (obsolète)

Version : 2.05.0

Aurora MySQL 2.05.0 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Versions Aurora MySQL actuellement prises en charge : 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* et 2.04.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL prise en charge sur Aurora MySQL 2.05.0. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers 2.04.6 ou Aurora MySQL 2.05.0. Vous ne pouvez pas mettre à niveau

directement un cluster Aurora MySQL 1.* existant vers 2.05.0. En revanche, vous pouvez restaurer son instantané vers Aurora MySQL 2.05.0.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Chine (Ningxia) [cn-northwest-1], Asie-Pacifique (Hong Kong) [ap-east-1], Europe (Stockholm) [eu-north-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs critiques :

- 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
- Résolution d'un problème selon lequel les événements du fichier de journaux binaires actuel sur le maître n'étaient pas répliqués sur le travailleur si la valeur du paramètre `sync_binlog` n'était pas définie sur 1.

Correctifs à priorité élevée :

- Il est vivement recommandé aux clients dont la taille de la base de données est proche de 64 tébioctets (TiO) de procéder à une mise à niveau vers cette version pour éviter les arrêts dus aux bogues de stabilité affectant les volumes proches de la limite de stockage Aurora.
- La valeur par défaut du paramètre `aurora_binlog_replication_max_yield_seconds` a été remplacée par zéro pour éviter une augmentation de l'écart de réplication en faveur de performances de requêtes de premier plan sur le fichier principal binlog.

## Intégration de correctifs de bogues MySQL.

- Bug #23054591 : PURGE BINARY LOGS TO lit l'intégralité du fichier binlog et provoque MySql le blocage

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.05.0 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.05.0 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 14/08/2020 (version 2.04.9) (obsolète)

Version : 2.04.9

Aurora MySQL 2.04.9 est disponible pour le grand public Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Asie-Pacifique (Hong Kong) [ap-east-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs à priorité élevée :

Correctifs critiques :

- 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

Améliorations de la disponibilité :

- Correction d'un problème qui pouvait provoquer un redémarrage ou un basculement de base de données en raison de l'exécution d'une commande `kill session`. Si vous rencontrez ce problème, contactez le AWS support pour activer ce correctif sur votre instance.
- Correction d'un problème qui provoquait un redémarrage de base de données lors de l'exécution d'une requête complexe impliquant des jointures multi-tables et une agrégation utilisant des tables intermédiaires en interne.
- Correction d'un problème qui provoquait le redémarrage de la base de données en raison d'une interruption de `DROP TABLE` sur plusieurs tables.

- Correction d'un problème provoquant un basculement de base de données pendant la récupération de base de données.
- Correction d'un redémarrage de base de données entraîné par un rapport incorrect de threads_running lorsque les journaux d'audit et de requêtes lentes sont activés.
- Correction d'un problème qui empêchait une commande `kill query` de rester bloquée pendant l'exécution.
- Correction d'une condition de concurrence dans le gestionnaire de verrous qui entraînait le redémarrage de la base de données ou le basculement pendant la restauration d'une transaction.
- Correction d'un problème qui entraînait le redémarrage ou le basculement de la base de données lorsque plusieurs connexions tentaient de mettre à jour la même table avec un index de recherche en texte intégral.
- Correction d'un problème qui pouvait provoquer un blocage lors de la purge d'un index entraînant un basculement ou un redémarrage.

Améliorations générales :

- Correction de problèmes qui pouvaient entraîner l'utilisation de données provenant d'une transaction non validée par les requêtes sur les réplicas en lecture. Ce problème est limité aux transactions qui sont démarrées immédiatement après un redémarrage de la base de données.
- Correction d'un problème rencontré lors de `INPLACE ALTER TABLE` pour une table avec des déclencheurs définis et lorsque le DDL ne contenait pas de clause `RENAME`.
- Correction d'un problème qui entraînait un clonage plus long sur certains clusters de bases de données avec une charge d'écriture élevée.
- Correction d'un problème rencontré lors d'une mise à niveau lorsqu'une table partitionnée comportait des espaces incorporés dans le nom.
- Correction d'un problème dans lequel le réplica en lecture pouvait voir temporairement les résultats partiels d'une transaction récemment validée sur le rédacteur.
- Correction d'un problème au cours duquel les requêtes sur un réplica en lecture sur une table FTS pouvaient produire des résultats obsolètes. Cela ne se produit que lorsque la requête FTS sur le réplica en lecture suit de près une requête sur `INFORMATION_SCHEMA.INNODB_SYS_TABLES` la même table FTS sur le writer.
- Correction d'un problème qui entraînait une restauration lente du cluster de bases de données Aurora 1.x contenant des index FTS (Full-Text Search) vers un cluster de bases de données Aurora 2.x.

- Longueur maximale autorisée étendue jusqu'à 2 000 pour les paramètres globaux `server_audit_incl_users` et `server_audit_excl_users`.
- Correction d'un problème au cours duquel la restauration d'Aurora 1.x vers Aurora 2.x pouvait prendre plus de temps.
- Correction d'un problème dans lequel un appel `lambda_async` via une procédure stockée ne fonctionne pas avec Unicode.
- Correction d'un problème rencontré lorsqu'un index spatial ne gérait pas correctement une colonne de géométrie hors enregistrement.
- Correction d'un problème qui pouvait provoquer l'échec d'une requête sur une instance de base de données de lecteur avec une erreur `InternalFailureException` et le message « Opération terminée (erreur interne) ».

## Intégration de correctifs de bogues MySQL.

- Bug #23070734, Bug #80060 : la fonction TRUNCATE simultanée TABLEs provoque des blocages
- Bogue n° 23103937 : PS_TRUNCATE_ALL_TABLES() NE FONCTIONNE PAS EN MODE SUPER_READ_ONLY
- Bogue n° #22551677 : lorsque vous mettez le serveur hors connexion, une condition de concurrence dans le schéma de performances peut entraîner la sortie du serveur.
- Bogue n° 27082268 : synchronisation FTS non valide.
- BOGUE n° 12589870 : correction d'un problème qui provoquait un redémarrage avec une instruction multi-requête lorsque le cache de requête est activé.
- Bogue n° 26402045 : certains cas de matérialisation de sous-requête peuvent provoquer la sortie d'un serveur. Ces requêtes produisent maintenant une erreur suggérant que la matérialisation doit être désactivée.
- Bogue n° 18898433 : les requêtes avec de nombreuses jointures gauche étaient lentes si la mise en mémoire tampon de jointure était utilisée (par exemple, en utilisant l'algorithme de boucle imbriquée par bloc).
- Bogue n° 25222337 : un nom de champ de colonne virtuelle NULL dans un index virtuel a provoqué une sortie du serveur lors d'une comparaison de noms de champ qui se produit lors du remplissage de colonnes virtuelles affectées par une contrainte de clé étrangère. (https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131)

- Bogue n° 25053286 : l'exécution d'une procédure stockée contenant une requête qui a accédé à une vue pouvait allouer de la mémoire qui n'était pas libérée jusqu'à la fin de la session. (https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d)
- Bug #25586773 : L'exécution d'une procédure stockée contenant une instruction qui a créé une table à partir du contenu de certaines instructions SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) peut entraîner une fuite de mémoire. (https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1)
- Bogue n° 26666274 : BOUCLE INFINITE DANS LE CONTAINER DE MÉMOIRE TAMPON DE SCHÉMA DE PERFORMANCE.
- Bogue n° 23550835, Bogue n° 23298025, Bogue n° 81464 : une table SELECT Performance Schema lorsqu'un tampon interne était plein pouvait provoquer la sortie d'un serveur.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.04.9 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.04.9 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 20/11/2019 (version 2.04.8) (obsolète)

Version : 2.04.8

Aurora MySQL 2.04.8 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer l'instantané d'une versions Aurora MySQL 2.* en Aurora MySQL 2.04.8. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.04.8.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Asie-Pacifique (Hong Kong) [ap-east-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Améliorations des réplicas en lecture :
  - Trafic réseau réduit depuis l'instance en écriture en transmettant efficacement les données aux instances du lecteur au sein du cluster de bases de données Aurora. Cette amélioration est activée par défaut, car elle aide à empêcher les réplicas de prendre du retard et de redémarrer. Le paramètre de cette fonctionnalité est `aurora_enable_repl_bin_log_filtering`.
  - Trafic réseau réduit depuis le dispositif d'écriture vers les instances du dispositif de lecture au sein du cluster de bases de données Aurora utilisant la compression. Cette amélioration est activée par défaut pour les classes d'instance 8xlarge et 16xlarge uniquement, car ces instances peuvent tolérer une surcharge d'UC additionnelle pour la compression. Le paramètre de cette fonctionnalité est `aurora_enable_replica_log_compression`.

Correctifs à priorité élevée :

- Gestion de la mémoire améliorée dans l'instance en écriture Aurora qui empêche le redémarrage du dispositif d'écriture suite à un dépassement de la mémoire pendant une charge de travail intensive en présence d'instances en lecture au sein du cluster de bases de données Aurora.
- Correction d'une condition non déterministe dans le planificateur qui entraîne le redémarrage du moteur tout en accédant simultanément à l'objet de schéma de performances.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.04.8 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.04.8 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 14/11/2019 (version 2.04.7) (obsolète)

Version : 2.04.7

Aurora MySQL 2.04.7 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer un instantané à partir d'une version Aurora MySQL prise en charge sur Aurora MySQL 2.04.7. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 2.* existants vers Aurora MySQL 2.04.7. Vous ne pouvez pas mettre à niveau directement un cluster Aurora MySQL 1.* existant vers 2.04.7. En revanche, vous pouvez restaurer son instantané sur Aurora MySQL 2.04.7.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de ou de l'API RDS. AWS CLI

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Asie-Pacifique (Hong Kong) [ap-east-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs à priorité élevée :

Gestion des connexions

- La disponibilité des bases de données a été améliorée afin de mieux répondre à l'augmentation du nombre de connexions clients lors de l'exécution d'une ou de plusieurs connexions DDLs. La gestion s'effectue en créant temporairement des threads supplémentaires si nécessaire. La mise à niveau est conseillée si la base de données ne répond plus suite à un pic dans les connexions, parallèlement au traitement des DDL.
- Correction d'un problème qui entraînait une valeur incorrecte pour la variable d'état globale `Threads_running`.

Redémarrage du moteur

- Correction du problème d'indisponibilité prolongée lors du redémarrage du moteur. La correction concerne un problème d'initialisation du pool de mémoires tampon. Ce problème se produit rarement, mais peut éventuellement impacter toute version prise en charge.

Correctifs de stabilité générale :

- Les améliorations apportées à l'accès des requêtes aux données non mises en cache peuvent être plus lentes qu'à l'ordinaire. Les clients confrontés à une latence en lecture anormalement élevé alors qu'ils accèdent aux données mises en cache sont invités à procéder à une mise à niveau.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.04.7 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.04.7 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe

- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 19/09/2019 (version 2.04.6) (obsolète)

Version : 2.04.6

Aurora MySQL 2.04.6 est généralement disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous avez la possibilité de mettre à niveau les clusters de base de données Aurora MySQL 2.* existants vers Aurora MySQL 2.04.6. Nous n'autorisons pas les mises à niveau en place des clusters Aurora MySQL 1.* Cette restriction sera levée dans une version ultérieure d'Aurora MySQL 2.*. Vous pouvez restaurer les instantanés Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* et 2.04.* dans Aurora MySQL 2.04.6.

Pour utiliser une ancienne version d'Aurora MySQL, vous pouvez créer de nouveaux clusters de bases de données en spécifiant la version du AWS Management Console moteur via l'API AWS CLI, ou Amazon RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : Europe (Londres) [eu-west-2] AWS GovCloud , (US-Est) [-1], (US-Ouest) us-gov-east [-1] AWS GovCloud , Chine (Ningxia) us-gov-west [cn-northwest-1] et Asie-Pacifique (Hong Kong) [ap-east-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Résolution d'un problème selon lequel les événements du fichier de journaux binaires actuel sur le maître n'étaient pas répliqués sur le travailleur si la valeur du paramètre `sync_binlog` n'était pas définie sur 1.
- La valeur par défaut du paramètre `aurora_binlog_replication_max_yield_seconds` a été remplacée par zéro pour éviter une augmentation de l'écart de réplication en faveur de performances de requêtes de premier plan sur le fichier principal binlog.

## Intégration de correctifs de bogues MySQL.

- Bug #23054591 : PURGE BINARY LOGS TO lit l'intégralité du fichier binlog et provoque MySql le blocage

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.04.6 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.04.6 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication

- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 08/07/2019 (version 2.04.5) (obsolète)

Version : 2.04.5

Aurora MySQL 2.04.5 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous avez la possibilité de mettre à niveau les clusters de base de données Aurora MySQL 2.* existants vers Aurora MySQL 2.04.5. Nous n'autorisons pas les mises à niveau en place des clusters Aurora MySQL 1.* Cette restriction sera levée dans une version ultérieure d'Aurora MySQL 2.*. Vous pouvez restaurer les instantanés Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* et 2.04.* dans Aurora MySQL 2.04.5.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

- CVE-2016-3518

Correctifs généraux :

- Résolution d'une situation de compétition lors de la croissance du volume de stockage qui entraînait le redémarrage de la base de données.
- Résolution d'une défaillance de communication interne pendant l'ouverture du volume qui entraînait le redémarrage de la base de données.
- Ajout de la prise en charge de la récupération de DDL pour `ALTER TABLE ALGORITHM=INPLACE` sur les tables partitionnées.
- Résolution d'un problème avec la récupération de DDL de `ALTER TABLE ALGORITHM=COPY` qui entraînait le redémarrage de la base de données.
- Amélioration de la stabilité de réplica d'Aurora sous une lourde charge de travail de suppression sur l'enregistreur.
- Résolution d'un redémarrage de la base de données causé par un interblocage entre le thread qui effectue la synchronisation de l'index en texte intégral et le thread qui procède à l'expulsion de la table de recherche en texte intégral du cache de dictionnaire.
- Résolution d'un problème de stabilité sur le travailleur de journaux binaires au cours de la réplication DDL lorsque la connexion au maître des journaux binaires est instable.
- Correction d'un out-of-memory problème dans le code de recherche en texte intégral qui provoquait le redémarrage de la base de données.
- Résolution d'un problème sur l'enregistreur Aurora qui entraînait son redémarrage lors de l'utilisation de l'intégralité du volume de 64 tébioctets (TiO).
- Résolution d'une situation de compétition dans la fonction de schéma de performance qui entraînait le redémarrage de la base de données.
- Résolution d'un problème qui interrompait les connexions lors du traitement d'une erreur de gestion du protocole réseau.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.04.5 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.04.5 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 29/05/2019 (version 2.04.4) (obsolète)

Version : 2.04.4

Aurora MySQL 2.04.4 est généralement disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Lorsque vous créez un cluster de bases de données Aurora MySQL (y compris lorsque vous restaurez un instantané) vous avez le choix entre la compatibilité avec MySQL 5.7 ou avec MySQL 5.6. Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* ou la restauration des clusters Aurora MySQL 1.* à partir d'une sauvegarde Amazon S3 vers Aurora MySQL 2.04.4. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Vous pouvez restaurer les instantanés Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* et 2.04.* dans Aurora MySQL 2.04.4.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (US-West) [us-gov-west-1], Europe (Stockholm) [eu-north-1], Chine (Ningxia) [cn-northwest-1] et Asie-Pacifique (Hong Kong) [ap-east-1]. AWS Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Résolution d'un problème qui pouvait entraîner des échecs lors du chargement de données vers Aurora à partir de S3.
- Résolution d'un problème qui pouvait entraîner des échecs lors du chargement de données à partir d'Aurora vers S3.
- Résolution d'un problème qui interrompait les connexions lors du traitement d'une erreur de gestion du protocole réseau.
- Résolution d'un problème qui pouvait entraîner un blocage lors du traitement de tables partitionnées.
- Résolution d'un problème rendant indisponible la fonction Performance Insights dans certaines régions.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.04.4 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.04.4 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 09/05/2019 (version 2.04.3) (obsolète)

Version : 2.04.3

Aurora MySQL 2.04.3 est en disponibilité générale. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Lorsque vous créez un cluster de bases de données Aurora MySQL (y compris lorsque vous restaurez un instantané) vous avez le choix entre la compatibilité avec MySQL 5.7 ou avec

MySQL 5.6. Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* ou la restauration des clusters Aurora MySQL 1.* à partir d'une sauvegarde Amazon S3 vers Aurora MySQL 2.04.3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Vous pouvez restaurer les instantanés Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* et 2.04.* dans Aurora MySQL 2.04.3.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (US-Ouest) [us-gov-west-1] et Chine (Ningxia) [cn-northwest-1]. AWS Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'un bogue dans la réplication des journaux binaires pouvant entraîner un problème sur les instances Aurora configurées en tant que travailleur des journaux binaires.
- Correction d'une out-of-memory condition lors de la gestion de routines stockées de grande taille.
- Correction d'une erreur survenant lors de la gestion de certains types de commandes `ALTER TABLE`.
- Correction d'une erreur d'interruption des connexions en raison d'une erreur de gestion du protocole réseau.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.04.3 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.04.3 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source

- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 02/05/2019 (version 2.04.2) (obsolète)

Version : 2.04.2

Aurora MySQL 2.04.2 est en disponibilité générale. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Lorsque vous créez un cluster de bases de données Aurora MySQL (y compris lorsque vous restaurez un instantané) vous avez le choix entre la compatibilité avec MySQL 5.7 ou avec MySQL 5.6. Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* ou la restauration des clusters Aurora MySQL 1.* à partir d'une sauvegarde Amazon S3 vers Aurora MySQL 2.04.2. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Vous pouvez restaurer les instantanés Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.0 et 2.04.1 dans Aurora MySQL 2.04.2.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (US-Ouest) [us-gov-west-1] et Chine (Ningxia) [cn-northwest-1]. AWS Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> Pour plus d'informations sur la mise à niveau de votre cluster de base de données Aurora MySQL, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Ajout de la prise en charge de la réplication des journaux binaires SSL à l'aide de certificats personnalisés. Pour plus d'informations sur l'utilisation de la réplication des journaux binaires SSL dans Aurora MySQL, veuillez consulter mysql_rds_import_binlog_ssl_material.
- Correction d'un interblocage sur l'instance principale Aurora qui se produit lorsqu'une table avec un index Recherche en texte intégral est optimisée.
- Correction d'un problème sur les réplicas Aurora dont la performance de certaines requêtes à l'aide de `SELECT(*)` pourrait être affectée sur les tables comportant des index secondaires.
- Correction d'une condition entraînant l'envoi d'une erreur 1032.
- Amélioration de la stabilité des réplicas Aurora par la correction de plusieurs interblocages.

## Intégration de correctifs de bogues MySQL.

- Bogue n°24829050 - INDEX_MERGE_INTERSECTION OPTIMIZATION CAUSES WRONG QUERY RESULTS

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.04.2 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.04.2 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 25/03/2019 (version 2.04.1) (obsolète)

Version : 2.04.1

Aurora MySQL 2.04.1 est en disponibilité générale. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Lorsque vous créez un cluster de bases de données Aurora MySQL (y compris lorsque vous restaurez un instantané) vous avez le choix entre la compatibilité avec MySQL 5.7 ou avec MySQL 5.6. Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* ou la restauration des clusters Aurora MySQL 1.* à partir d'une sauvegarde Amazon S3 vers Aurora MySQL 2.04.1. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Vous pouvez restaurer des instantanés Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.0 dans Aurora MySQL 2.04.1.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans la région AWS GovCloud (US-Ouest) [us-gov-west-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Résolution d'un problème où un instantané Aurora MySQL 5.6 pour les versions antérieures à 1.16 ne pouvait pas être restauré avec le dernier cluster Aurora MySQL 5.7.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 25/03/2019 (version 2.04.0) (obsolète)

Version : 2.04

Aurora MySQL 2.04 est en disponibilité générale. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Lorsque vous créez un cluster de bases de données Aurora MySQL (y compris lorsque vous restaurez un instantané) vous avez le choix entre la compatibilité avec MySQL 5.7 ou avec MySQL 5.6. Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* ou la restauration des clusters Aurora MySQL 1.* à partir d'une sauvegarde Amazon S3 vers Aurora MySQL 2.04.0. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Vous pouvez restaurer les instantanés Aurora MySQL 1.19.*, 2.01.*, 2.02.* et 2.03.* dans Aurora MySQL 2.04.0. Vous ne pouvez pas restaurer les instantanés Aurora MySQL 1.14.* ou version antérieure, 1.15.*, 1.16.*, 1.17.*, 1.18.* dans Aurora MySQL 2.04.0. Cette restriction ne s'applique pas à la version Aurora MySQL 2.04.1.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans la région AWS GovCloud (US-Ouest) [us-gov-west-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Prise en charge de la réplication basée sur des identifiants de transaction globaux (GTID). Pour plus d'informations sur l'utilisation de la réplication basée sur GTID avec Aurora MySQL, consultez Utilisation de la réplication basée sur GTID pour Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Résolution d'un problème où un réplica Aurora lève une erreur `Running in read-only mode` de manière incorrecte lorsqu'une instruction qui supprime ou met à jour des lignes dans une table temporaire contient une sous-requête InnoDB.

## Intégration de correctifs de bogues MySQL.

- Bogue n°26225783 : MYSQL CRASH ON CREATE TABLE (REPRODUCEABLE) -> INNODB: ALONG SEMAPHORE WAIT.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL Version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL Version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Cette version d'Aurora MySQL est compatible réseau avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Cette version d'Aurora MySQL ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB

- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

# Mises à jour du moteur de base de données Aurora MySQL du 07/02/2019 (version 2.03.4) (obsolète)

Version : 2.03.4

Aurora MySQL 2.03.4 est en disponibilité générale. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL (y compris en restaurant un instantané), vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.03.4 ni la restauration vers Aurora MySQL 2.03.4 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Support du classement UTF8 MB4 Unicode 9.0 sensible aux accents et insensible aux majuscules et minuscules,. `utf8mb4_0900_as_ci`

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.03.4 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.03.4 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL du 18/01/2019 (version 2.03.3) (obsolète)

Version : 2.03.3

Aurora MySQL 2.03.3 est en disponibilité générale. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL (y compris en restaurant un instantané), vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.03.3 ni la restauration vers Aurora MySQL 2.03.3 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs CVE

- CVE-2016-5436

Correctifs critiques :

- Résolution d'un problème ou un réplica Aurora peut devenir interbloqué lors de l'exécution d'une analyse en arrière sur un index.
- Résolution d'un problème où un réplica Aurora peut redémarrer lorsque l'instance principale Aurora exécute des opérations DDL sur place sur des tables partitionnées.
- Résolution d'un problème où un réplica Aurora peut redémarrer pendant l'invalidation du cache de requêtes après une opération DDL sur l'instance principale Aurora.
- Résolution d'un problème où un réplica Aurora peut redémarrer pendant une requête `SELECT` sur une table alors que l'instance principale Aurora exécute une troncature sur cette table.

- Résolution d'un problème de résultat incorrect avec les tables temporaires MyISAM dont seules les colonnes indexées sont accessibles.
- Résolution d'un problème de journaux lents générant des valeurs élevées incorrectes pour `query_time` et `lock_time` régulièrement après environ 40 000 requêtes.
- Résolution d'un problème dans lequel un schéma nommé « tmp » pouvait entraîner le blocage de la migration de RDS for MySQL vers Aurora MySQL.
- Résolution d'un problème où le journal d'audit peut comporter des événements manquants lors de la rotation des journaux.
- Résolution d'un problème où l'instance principale Aurora restaurée à partir d'un instantané Aurora 5.6 peut redémarrer lorsque la fonction Fast DDL en mode Lab est activée.
- Résolution d'un problème où l'utilisation de l'IUC est intégralement liée au thread stats du dictionnaire.
- Résolution d'un problème où un réplica Aurora peut redémarrer lors de l'exécution d'une instruction `CHECK TABLE`.

## Intégration de correctifs de bogues MySQL.

- Bogue n°25361251 : INCORRECT BEHAVIOR WITH INSERT ON DUPLICATE KEY IN SP
- Bogue n°26734162 : INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE
- Bogue n°27460607 : INCORRECT BEHAVIOR OF IODKU WHEN INSERT SELECT's SOURCE TABLE IS EMPTY
- Une requête utilisant une clause `DISTINCT` ou `GROUP BY` peut renvoyer des résultats incorrects. (MySQL 5.7 bogue n°79591, bogue n°22343910)
- A `DELETE` from joined tables using a derived table in the `WHERE` clause fails with error 1093 (Bogue n° 23074801).
- GCOLS: INCORRECT BEHAVIOR WITH CHARSET CHANGES (Bogue n°25287633).

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.03.3 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.03.3 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête

- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL du 09/01/2019 (version 2.03.2) (obsolète)

Version : 2.03.2

Aurora MySQL 2.03.2 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL (y compris en restaurant un instantané), vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.03.2 ni la restauration vers Aurora MySQL 2.03.2 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Sélecteur de version Aurora – Depuis la version Aurora MySQL 2.03.2, vous avez le choix entre plusieurs versions d'Aurora compatibles avec MySQL 5.7 dans l' AWS Management Console. Pour plus d'informations, consultez Vérification ou spécification de versions du moteur Aurora MySQL via AWS dans le Guide de l'utilisateur Amazon Aurora.

Correctifs critiques :

- CVE-2016-3495

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.03.2 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une

implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.03.2 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL du 24/10/2018 (version 2.03.1) (obsolète)

Version : 2.03.1

Aurora MySQL 2.03.1 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6. Lorsque vous restaurez un instantané compatible avec MySQL 5.6, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.* et 2.03 dans Aurora MySQL 2.03.1.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.03.1 ni la restauration vers Aurora MySQL 2.03.1 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

## Améliorations

- Correction d'un problème de redémarrage possible de l'enregistreur Aurora lors de l'exécution de la détection de blocage de transaction.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.03.1 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.03.1 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL du 11/10/2018 (version 2.03) (obsolète)

Version : 2.03

Aurora MySQL 2.03 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6. Lorsque vous restaurez un instantané compatible avec MySQL 5.6, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.* et 2.02.* dans Aurora MySQL 2.03.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.03 ni la restauration vers Aurora MySQL 2.03 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Disponibilité du schéma de performances.
- Correction d'un problème à cause duquel les sessions zombie à l'état Supprimé peuvent consommer davantage d'UC.
- Correction d'un problème de verrou mort lorsqu'une transaction en lecture seule acquiert un verrouillage sur un enregistrement de l'enregistreur Aurora.
- Correction d'un problème à cause duquel le réplica Aurora sans charge de travail client peut avoir une utilisation élevée de l'UC.
- Plusieurs correctifs pour des problèmes pouvant entraîner le redémarrage du réplica Aurora ou de l'enregistreur Aurora.

- Ajout de la capacité d'ignorer la journalisation des diagnostics lorsque la limite de débit de disque est atteinte.
- Correction d'un problème de fuite de mémoire lorsque le journal binaire est activé sur l'enregistreur Aurora.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- UNE ANALYSE EN ORDRE INVERSE SUR UNE TABLE PARTITIONNÉE DONNE UN CLASSEMENT ICP PAR DESCRIPTION (bogue n° 24929748).
- JSON_OBJECT CRÉE UN CODE JSON NON VALIDE (bogue n° 26867509).
- L'INSERTION D'UN GROS VOLUME DE DONNÉES JSON PREND UNE DURÉE INHABITUELLE (bogue n° 22843444).
- LES TABLES PARTITIONNÉES UTILISENT DAVANTAGE DE MÉMOIRE DANS LA VERSION 5.7 QUE DANS LA VERSION 5.6 (bogue n° 25080442).

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.03 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.03 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL du 08/10/2018 (version 2.02.5) (obsolète)

Version : 2.02.5

Aurora MySQL 2.02.5 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6. Lorsque vous restaurez un

instantané compatible avec MySQL 5.6, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.* et 2.02.* dans Aurora MySQL 2.02.5. Vous pouvez également effectuer une mise à niveau sur place d'Aurora MySQL 2.01.* ou 2.02.* vers Aurora MySQL 2.02.5.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.02.5 ni la restauration vers Aurora MySQL 2.02.5 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Le schéma de performances est désactivé pour cette version d'Aurora MySQL 5.7. Faites une mise à niveau vers Aurora 2.03 pour avoir la prise en charge du schéma de performances.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'un problème à cause duquel un réplica Aurora peut redémarrer lorsqu'il effectue une analyse en ordre inverse sur une table.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.02.5 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.02.5 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL du 21/09/2018 (version 2.02.4) (obsolète)

Version : 2.02.4

Aurora MySQL 2.02.4 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6. Lorsque vous restaurez un instantané compatible avec MySQL 5.6, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.* et 2.02.* dans Aurora MySQL 2.02.4. Vous pouvez également effectuer une mise à niveau sur place d'Aurora MySQL 2.01.* ou 2.02.* vers Aurora MySQL 2.02.4.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.02.4 ni la restauration vers Aurora MySQL 2.02.4 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Le schéma de performances est désactivé pour cette version d'Aurora MySQL 5.7. Faites une mise à niveau vers Aurora 2.03 pour avoir la prise en charge du schéma de performances.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'un problème de stabilité lié aux index de recherche en texte intégral sur les tables restaurées à partir d'un instantané Aurora MySQL 5.6.

## Intégration de correctifs de bogues de l'édition MySQL 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

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.

- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.02.4 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.02.4 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`

- Protocole X

# Mises à jour du moteur de base de données Aurora MySQL du 23/08/2018 (version 2.02.3) (obsolète)

Version : 2.02.3

Aurora MySQL 2.02.3 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6. Lorsque vous restaurez un instantané compatible avec MySQL 5.6, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 2.01.* et 2.02.* dans Aurora MySQL 2.02.3. Vous pouvez également effectuer une mise à niveau sur place d'Aurora MySQL 2.01.* ou 2.02.* vers Aurora MySQL 2.02.3.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.02.3 ni la restauration vers Aurora MySQL 2.02.3 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Le schéma de performances est désactivé pour cette version d'Aurora MySQL 5.7. Faites une mise à niveau vers Aurora 2.03 pour avoir la prise en charge du schéma de performances.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Comparaison avec Aurora MySQL Version 1

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL version 1 (compatible avec MySQL 5.6), mais ne sont pas actuellement prises en charge dans Aurora MySQL version 2 (compatible avec MySQL 5.7).

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

Actuellement, Aurora MySQL 2.01 ne prend pas en charge les fonctions ajoutées dans Aurora MySQL version 1.16 et versions ultérieures. Pour plus d'informations sur Aurora MySQL version 1.16, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.02.3 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.02.3 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

## Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x

- Le nom de moteur d'Aurora MySQL 2.x est `aurora-mysql` ; le nom de moteur d'Aurora MySQL 1.x reste `aurora`.
- Le groupe de paramètres par défaut d'Aurora MySQL 2.x est `default.aurora-mysql5.7` ; le groupe de paramètres par défaut d'Aurora MySQL 1.x reste `default.aurora5.6`.
- Le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 2.x est `aurora-mysql5.7` ; le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 1.x reste `aurora5.6`.

Consultez la documentation Aurora pour voir l'ensemble des commandes de CLI et des différences entre Aurora MySQL 2.x et Aurora MySQL 1.x.

## Améliorations

- Correction d'un problème impliquant le redémarrage d'un réplica Aurora lors de l'utilisation des restaurations de curseur optimisé pendant la lecture des enregistrements.
- Mise à jour de la valeur par défaut du paramètre `innodb_stats_persistent_sample_pages` à 128 pour améliorer les statistiques d'index.

- Correction d'un problème de redémarrage possible d'un réplica Aurora lorsqu'il accède à une petite table qui est modifiée simultanément sur l'instance principale Aurora.
- Correction d'`ANALYZE TABLE` pour arrêter de vider le cache de définition de table.
- Correction d'un problème de redémarrage possible de l'instance principale Aurora ou d'un réplica Aurora lors de la conversion d'une requête de point géospatial en plage de recherche.

# Mises à jour du moteur de base de données Aurora MySQL du 04/06/2018 (version 2.02.2) (obsolète)

Version : 2.02.2

Aurora MySQL 2.02.2 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6. Lorsque vous restaurez un instantané compatible avec MySQL 5.6, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17*, 2.01* et 2.02 dans Aurora MySQL 2.02.2. Vous pouvez également effectuer une mise à niveau sur place d'Aurora MySQL 2.01* ou 2.02 vers Aurora MySQL 2.02.2.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.* vers Aurora MySQL 2.02.2 ni la restauration vers Aurora MySQL 2.02.2 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.*.

Le schéma de performances est désactivé pour cette version d'Aurora MySQL 5.7. Faites une mise à niveau vers Aurora 2.03 pour avoir la prise en charge du schéma de performances.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs critiques :

- CVE-2016-3486

## Comparaison avec Aurora MySQL 5.6

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL 5.6, mais pas dans Aurora MySQL 5.7.

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

Actuellement, Aurora MySQL 2.01 ne prend pas en charge les fonctions ajoutées dans Aurora MySQL version 1.16 et versions ultérieures. Pour plus d'informations sur Aurora MySQL version 1.16, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.02.2 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.02.2 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

## Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x

- Le nom de moteur d'Aurora MySQL 2.x est `aurora-mysql` ; le nom de moteur d'Aurora MySQL 1.x reste `aurora`.
- Le groupe de paramètres par défaut d'Aurora MySQL 2.x est `default.aurora-mysql5.7` ; le groupe de paramètres par défaut d'Aurora MySQL 1.x reste `default.aurora5.6`.
- Le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 2.x est `aurora-mysql5.7` ; le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 1.x reste `aurora5.6`.

Consultez la documentation Aurora pour voir l'ensemble des commandes de CLI et des différences entre Aurora MySQL 2.x et Aurora MySQL 1.x.

### Améliorations

- Correction d'un problème où un enregistreur Aurora redémarre parfois lors du suivi de la progression d'Aurora Replica.
- Correction d'un problème à cause duquel un réplica Aurora redémarre ou génère une erreur lorsqu'une table partitionnée fait l'objet d'un accès après l'exécution d'instructions de création ou d'abandon d'index sur la table de l'enregistreur Aurora.
- Correction d'un problème en raison duquel une table d'une réplique d'Aurora était inaccessible lorsqu'elle appliquait les modifications provoquées par l'exécution des instructions de ADD/DROP colonne de table ALTER sur l'Aurora Writer.

## Mises à jour du moteur de base de données Aurora MySQL du 03/05/2018 (version 2.02) (obsolète)

Version : 2.02

Aurora MySQL 2.02 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6. Lorsque vous restaurez un instantané compatible avec MySQL 5.6, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17* et 2.01* dans Aurora MySQL 2.02. Vous pouvez également effectuer une mise à niveau sur place d'Aurora MySQL 2.01* vers Aurora MySQL 2.02.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.x vers Aurora MySQL 2.02, ni la restauration vers Aurora MySQL 2.02 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.x.

Le schéma de performances est désactivé pour cette version d'Aurora MySQL 5.7. Faites une mise à niveau vers Aurora 2.03 pour avoir la prise en charge du schéma de performances.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Comparaison avec Aurora MySQL 5.6

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL 5.6, mais pas dans Aurora MySQL 5.7.

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

Actuellement, Aurora MySQL 2.01 ne prend pas en charge les fonctions ajoutées dans Aurora MySQL version 1.16 et versions ultérieures. Pour plus d'informations sur Aurora MySQL version 1.16, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.02 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.02 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

## Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x

- Le nom de moteur d'Aurora MySQL 2.x est `aurora-mysql` ; le nom de moteur d'Aurora MySQL 1.x reste `aurora`.
- Le groupe de paramètres par défaut d'Aurora MySQL 2.x est `default.aurora-mysql5.7` ; le groupe de paramètres par défaut d'Aurora MySQL 1.x reste `default.aurora5.6`.
- Le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 2.x est `aurora-mysql5.7` ; le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 1.x reste `aurora5.6`.

Consultez la documentation Aurora pour voir l'ensemble des commandes de CLI et des différences entre Aurora MySQL 2.x et Aurora MySQL 1.x.

## Améliorations

- Correction d'un problème où un enregistreur Aurora redémarre lors de l'exécution d'instructions INSERT et de l'exploitation de l'optimisation Insertion rapide.

- Correction d'un problème où un réplica Aurora redémarre lors de l'exécution d'instructions ALTER DATABASE sur le réplica Aurora.
- Correction d'un problème où un réplica Aurora redémarre lors de l'exécution de requêtes sur des tables qui viennent d'être supprimées sur l'enregistreur Aurora.
- Correction d'un problème où un réplica Aurora redémarre lors de la définition de `innodb_adaptive_hash_index` sur `OFF` sur le réplica Aurora.
- Correction d'un problème où un réplica Aurora redémarre lors de l'exécution de requêtes TRUNCATE TABLE sur l'enregistreur Aurora.
- Correction d'un problème où l'enregistreur Aurora se fige dans certaines circonstances lors de l'exécution d'instructions INSERT. Sur un cluster à plusieurs nœuds, cela peut provoquer un basculement.
- Correction d'une fuite de mémoire associée à la définition de variables de session.
- Correction d'un problème où l'enregistreur Aurora se fige dans certaines circonstances liées à la purge d'annulation pour des tables contenant des colonnes générées.
- Correction d'un problème où l'enregistreur Aurora redémarre parfois lorsque la journalisation binaire est activée.

### Intégration de correctifs de bogues MySQL.

- La jointure gauche retourne des résultats incorrects du côté extérieur (Bogue n° 22833364).

## Mises à jour du moteur de base de données Aurora MySQL du 13/03/2018 (version 2.01.1) (obsolète)

Version : 2.01.1

Aurora MySQL 2.01.1 est disponible. Les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6. Lorsque vous restaurez un instantané compatible avec MySQL 5.6, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14*, 1.15*, 1.16* et 1.17* dans Aurora MySQL 2.01.1.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.x vers Aurora MySQL 2.01.1, ni la restauration vers Aurora MySQL 2.01.1 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.x.

Le schéma de performances est désactivé pour cette version d'Aurora MySQL 5.7. Faites une mise à niveau vers Aurora 2.03 pour avoir la prise en charge du schéma de performances.

## Comparaison avec Aurora MySQL 5.6

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL 5.6, mais pas dans Aurora MySQL 5.7.

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

Actuellement, Aurora MySQL 2.01.1 ne prend pas en charge les fonctions ajoutées dans Aurora MySQL version 1.16 et versions ultérieures. Pour plus d'informations sur Aurora MySQL version 1.16, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.01.1 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.01.1 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB
- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

## Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x

- Le nom de moteur d'Aurora MySQL 2.x est `aurora-mysql` ; le nom de moteur d'Aurora MySQL 1.x reste `aurora`.
- Le groupe de paramètres par défaut d'Aurora MySQL 2.x est `default.aurora-mysql5.7` ; le groupe de paramètres par défaut d'Aurora MySQL 1.x reste `default.aurora5.6`.
- Le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 2.x est `aurora-mysql5.7` ; le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 1.x reste `aurora5.6`.

Consultez la documentation Aurora pour voir l'ensemble des commandes de CLI et des différences entre Aurora MySQL 2.x et Aurora MySQL 1.x.

### Améliorations

- Correction du problème de restauration d'un instantané où les privilèges de base de données propres à Aurora n'étaient pas créés correctement quand un instantané compatible avec MySQL 5.6 était restauré avec la compatibilité MySQL 5.7.
- Ajout de la prise en charge des restaurations d'instantané 1.17.

## Mises à jour du moteur de base de données Aurora MySQL du 06/02/2018 (version 2.01) (obsolète)

Version : 2.01

Aurora MySQL 2.01 est disponible. À l'avenir, les versions 2.x d'Aurora MySQL seront compatibles avec MySQL 5.7 et les versions 1.x d'Aurora MySQL seront compatibles avec MySQL 5.6.

Lorsque vous créez un cluster de bases de données Aurora MySQL, y compris en restaurant un instantané, vous avez le choix entre la compatibilité avec MySQL 5.7 et la compatibilité avec MySQL 5.6.

Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14*, 1.15* et 1.16* dans Aurora MySQL 2.01.

Nous n'autorisons pas la mise à niveau sur place des clusters Aurora MySQL 1.x vers Aurora MySQL 2.01, ni la restauration vers Aurora MySQL 2.01 à partir d'une sauvegarde Amazon S3. Nous envisageons de supprimer ces restrictions dans une prochaine version d'Aurora MySQL 2.x.

Le schéma de performances est désactivé pour cette version d'Aurora MySQL 5.7. Faites une mise à niveau vers Aurora 2.03 pour avoir la prise en charge du schéma de performances.

### Comparaison avec Aurora MySQL 5.6

Les fonctions Amazon Aurora MySQL suivantes sont prises en charge dans Aurora MySQL 5.6, mais pas dans Aurora MySQL 5.7.

- Lecture anticipée asynchrone des clés (AKP). Pour plus d'informations, consultez Optimisation des requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.
- Jointures par hachage. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Fonctions natives pour invoquer AWS Lambda des fonctions de manière synchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda avec une fonction native Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Traitement par lot des analyses. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).
- Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3. Pour plus d'informations, consultez Migration des données à partir de MySQL à l'aide d'un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

Actuellement, Aurora MySQL 2.01 ne prend pas en charge les fonctions ajoutées dans Aurora MySQL version 1.16 et versions ultérieures. Pour plus d'informations sur Aurora MySQL version 1.16, consultez Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète).

## Compatibilité avec MySQL 5.7

Aurora MySQL 2.01 est compatible avec MySQL 5.7 et inclut des fonctions telles que la prise en charge de JSON, les index spatiaux et les colonnes générées. Aurora MySQL utilise une implémentation native de l'indexation spatiale à l'aide de courbes en z pour offrir des performances d'écriture 20 fois meilleures et des performances de lecture 10 fois meilleures que MySQL 5.7 pour des ensembles de données spatiaux.

Aurora MySQL 2.01 ne prend actuellement pas en charge les fonctions MySQL 5.7 suivantes :

- Identifiants de transaction globaux (GTIDs). Aurora MySQL est compatible avec GTIDs les versions 2.04 et supérieures.
- plugin de réplication de groupe
- Augmentation de la taille de page
- Chargement du pool de mémoires tampons InnoDB au démarrage
- plugin d'analyse de texte intégral InnoDB

- Réplication multi-source
- Redimensionnement de pool de mémoires tampons en ligne
- plugin de validation de mot de passe
- plugins de réécriture de requête
- Filtrage de réplication
- Instruction SQL `CREATE TABLESPACE`
- Protocole X

## Différences de CLI entre Aurora MySQL 2.x et Aurora MySQL 1.x

- Le nom de moteur d'Aurora MySQL 2.x est `aurora-mysql` ; le nom de moteur d'Aurora MySQL 1.x reste `aurora`.
- Le groupe de paramètres par défaut d'Aurora MySQL 2.x est `default.aurora-mysql5.7` ; le groupe de paramètres par défaut d'Aurora MySQL 1.x reste `default.aurora5.6`.
- Le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 2.x est `aurora-mysql5.7` ; le nom de famille du groupe de paramètres de cluster de bases de données d'Aurora MySQL 1.x reste `aurora5.6`.

Consultez la documentation Aurora pour voir l'ensemble des commandes de CLI et des différences entre Aurora MySQL 2.x et Aurora MySQL 1.x.

# Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 1 (obsolète)

Les mises à jour du moteur de base de données Amazon Aurora version 1 sont les suivantes :

- Mises à jour du moteur de base de données Aurora MySQL du 30/09/2021 (version 1.23.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 28/06/2021 (version 1.23.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 18/03/2021 (version 1.23.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 24/11/2020 (version 1.23.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 02/09/2020 (version 1.23.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 03/06/2021 (version 1.22.5) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 04/02/2021 (version 1.22.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 19/11/2020 (version 1.22.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 05/03/2020 (version 1.22.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 23/12/2019 (version 1.22.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 1.22.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 1.21.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 05/03/2020 (version 1.20.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 11/11/2019 (version 1.20.0) (obsolète)

- Mises à jour du moteur de base de données Aurora MySQL du 05/03/2020 (version 1.19.6) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 19/09/2019 (version 1.19.5) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 05/06/2019 (version 1.19.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 09/05/2019 (version 1.19.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 07/02/2019 (version 1.19.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 20/09/2018 (version 1.18.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 05/03/2020 (version 1.17.9) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 17/01/2019 (version 1.17.8) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 08/10/2018 (version 1.17.7) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 06/09/2018 (version 1.17.6) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 14/08/2018.(version 1.17.5) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 07/08/2018 (version 1.17.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 05/06/2018 (version 1.17.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 27/04/2018 (version 1.17.2) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 23/03/2018 (version 1.17.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 13/03/2018 (version 1.17) (obsolète)

- Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 20/11/2017 (version 1.15.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL du 24/10/2017 (version 1.15) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 13/03/2018 (version 1.14.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 22/09/2017 (version 1.14.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 07/08/2017 (version 1.14) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 15/05/2017 (version 1.13) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 05/04/2017 (version 1.12) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 23/02/2017 (version 1.11) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 12/01/2017 (version 1.10.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 14/12/2016 (version 1.10) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 10/11/2016 (versions 1.9.0, 1.9.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 26/10/2016 (version 1.8.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 18/10/2016 (version 1.8) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 20/09/2016 (version 1.7.1) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 30/08/2016 (version 1.7.0) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 01/06/2016 (version 1.6.5) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 06/04/2016 (version 1.6) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 11/01/2016 (version 1.5) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 03/12/2015 (version 1.4) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 16/10/2015 (versions 1.2, 1.3) (obsolète)
- Mises à jour du moteur de base de données Aurora MySQL : 24/08/2015 (version 1.1) (obsolète)

# Mises à jour du moteur de base de données Aurora MySQL du 30/09/2021 (version 1.23.4) (obsolète)

Version : 1.23.4

Aurora MySQL 1.23.4 est disponible. Les versions 2.* d'Aurora MySQL sont compatibles avec MySQL 5.7 et les versions 1.* d'Aurora MySQL sont compatibles avec MySQL 5.6.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Améliorations générales :

- Correction d'un problème qui pouvait entraîner une consommation élevée de l'UC sur les instances de lecture en raison de la journalisation excessive des messages d'information dans les fichiers journaux de diagnostic internes.

Correctifs à priorité élevée :

- CVE-2021-2307
- CVE-2021-2226
- CVE-2021-2160
- CVE-2021-2154
- CVE-2021-2060

- CVE-2021-2032
- CVE-2021-2001

# Mises à jour du moteur de base de données Aurora MySQL du 28/06/2021 (version 1.23.3) (obsolète)

Version : 1.23.3

Aurora MySQL 1.23.3 est disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Améliorations générales de la stabilité et de la disponibilité.

Correctifs de sécurité :

- CVE-2021-23841
- CVE-2021-3449
- CVE-2020-28196

# Mises à jour du moteur de base de données Aurora MySQL du 18/03/2021 (version 1.23.2) (obsolète)

Version : 1.23.2

Aurora MySQL 1.23.2 est disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs à priorité élevée :

- CVE-2020-14867
- CVE-2020-14812
- CVE-2020-14769
- CVE-2020-14765
- CVE-2020-14793
- CVE-2020-14672
- CVE-2020-1971
- CVE-2018-3143

Améliorations de la disponibilité :

- Correction d'un problème dans la fonction de redimensionnement dynamique qui pouvait entraîner le redémarrage des instances de base de données du lecteur.
- Correction d'un problème de basculement dû à une condition de concurrence dans l'instruction `RESET QUERY CACHE`.
- Correction d'un incident dans un appel de procédure stockée imbriquée avec cache de requête.
- Correction d'un problème afin d'empêcher le redémarrage répété de `mysqld` lors de la récupération à partir d'une troncature incomplète de tables partitionnées ou sous-partitionnées.
- Correction d'un problème qui pouvait provoquer l'échec d'une migration depuis le site ou RDS for MySQL vers Aurora MySQL.
- Correction d'une condition de concurrence rare dans laquelle la base de données pouvait redémarrer pendant la mise à l'échelle du volume de stockage.
- Correction d'un problème lié au gestionnaire de verrous, où une condition de concurrence pouvait provoquer le partage d'un verrou par deux transactions, provoquant le redémarrage de la base de données.
- Correction d'un problème lié à la gestion de la mémoire de verrouillage des transactions avec des transactions d'écriture longues entraînant un redémarrage de la base de données.
- Correction d'une condition de concurrence dans le gestionnaire de verrous qui entraînait le redémarrage de la base de données ou le basculement pendant la restauration d'une transaction.
- Correction d'un problème lors de la mise à niveau de la version 5.6 vers la version 5.7 lorsque la fonction Fast Online DDL était activée dans la table en mode Lab dans la version 5.6.
- Correction de plusieurs problèmes dans lesquels le moteur pouvait redémarrer pendant l'application de correctifs sans temps d'arrêt lors de la recherche d'un point de repos dans l'activité de la base de données pour l'application de correctifs.
- Correction de plusieurs problèmes liés aux redémarrages répétés dus à l'interruption des opérations DDL, telles que `DROP TRIGGER`, `ALTER TABLE` et en particulier `ALTER TABLE` qui modifie le type de partitionnement ou le nombre de partitions dans une table.
- Mise à jour de la valeur par défaut de `table_open_cache` sur les instances 16XL et 24XL afin d'éviter des redémarrages répétés et une utilisation élevée du processeur sur les classes d'instances volumineuses (R4/R5-16XL, R5-12XL, R5-24XL). Cela a eu un impact sur les versions 1.21.x et 1.22.x.
- Correction d'un problème qui provoquait l'arrêt d'un réplica de journal binaire avec une erreur `HA_ERR_KEY_NOT_FOUND`.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Réplication : pendant l'exécution d'une instruction `SHOW BINLOG EVENTS`, toute transaction parallèle a été bloquée. Le correctif garantit que le processus `SHOW BINLOG EVENTS` n'acquiert désormais un verrou que pendant la durée du calcul de la position finale du fichier ; par conséquent, les transactions parallèles ne sont pas bloquées pendant de longues durées. (Bogue n° 76618, Bogue n° 20928790)

## Mises à jour du moteur de base de données Aurora MySQL du 24/11/2020 (version 1.23.1) (obsolète)

Version : 1.23.1

Aurora MySQL 1.23.1 est disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- CVE-2020-14559
- CVE-2020-14539

Modifications incompatibles :

Cette version introduit une modification d'autorisation qui affecte le comportement de la commande `mysqldump`. Les utilisateurs doivent disposer du privilège `PROCESS` pour accéder à la table `INFORMATION_SCHEMA.FILES`. Pour exécuter la commande `mysqldump` sans aucune modification, accordez le privilège `PROCESS` à l'utilisateur de base de données auquel la commande `mysqldump` se connecte. Vous pouvez également exécuter la commande `mysqldump` avec l'option `--no-tablespaces`. Avec cette option, la sortie `mysqldump` n'inclut aucune instruction `CREATE LOGFILE GROUP` ou `CREATE TABLESPACE`. Dans ce cas, la commande `mysqldump` n'accède pas à la table `INFORMATION_SCHEMA.FILES` et vous n'avez pas besoin d'accorder l'autorisation `PROCESS`.

Améliorations de la disponibilité :

- Correction d'un problème entraînant le redémarrage répété d'une instance de lecteur Aurora dans un cluster secondaire de base de données globale exécutant 1.23.0.
- Correction d'un problème entraînant le redémarrage des réplicas d'une région secondaire de base de données globale lors de la mise à niveau vers la version 1.23.0 alors que le scripteur de la région principale était sur une version antérieure.
- Correction d'une fuite de mémoire dans la fonction de redimensionnement dynamique, introduite dans Aurora MySQL 1.23.0.
- Correction d'un problème pouvant entraîner le redémarrage du serveur pendant l'exécution d'une requête à l'aide de la fonction de requête parallèle.
- Correction d'un problème pouvant entraîner le blocage d'une séance client lorsque le moteur de base de données rencontrait une erreur lors de la lecture ou de l'écriture sur le réseau.

# Mises à jour du moteur de base de données Aurora MySQL du 02/09/2020 (version 1.23.0) (obsolète)

Version : 1.23.0

Aurora MySQL 1.23.0 est disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer l'instantané d'une base de données Aurora MySQL 1.* à la version Aurora MySQL 1.23.0.

> **Important**
>
> Les améliorations apportées au stockage Aurora dans cette version limitent les chemins de mise à niveau disponibles d'Aurora MySQL 1.23 à Aurora MySQL 2.*. Lorsque vous mettez à niveau un cluster d'Aurora MySQL version 1.23 vers une version 2.*, vous devez effectuer une mise à niveau vers Aurora MySQL version 2.09.0 ou ultérieure.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Vous pouvez désormais activer ou désactiver la requête parallèle pour un cluster existant en modifiant la valeur du paramètre de cluster de base de données `aurora_parallel_query`. Vous n'avez pas besoin d'utiliser `parallelquery` pour le paramètre `--engine-mode` lors de la création du cluster.

  La requête parallèle est désormais étendue et disponible dans toutes les régions où Aurora MySQL est disponible.

Un certain nombre d'autres améliorations de fonction et de modifications de procédure ont été apportées pour la mise à niveau et l'activation des requêtes parallèles dans un cluster Aurora. Pour plus d'informations, consultez Utilisation des requêtes parallèles pour Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

- Avec cette version, vous pouvez créer des instances de base de données Amazon Aurora MySQL avec jusqu'à 128 tébioctets (TiO) de stockage. La nouvelle limite de stockage est une augmentation par rapport à la limite précédente de 64 Tio. La taille de stockage de 128 Tio prend en charge des bases de données de taille plus importante. Cette capacité n'est pas prise en charge sur les petites tailles d'instance (db.t2 ou db.t3). Un espace disque logique unique ne peut pas dépasser 64 Tio en raison des limitations InnoDB avec une taille de page de 16 Ko.

  Aurora vous avertit lorsque la taille du volume du cluster est proche de 128 Tio, afin que vous puissiez prendre des mesures avant d'atteindre la limite de taille. Les alertes apparaissent dans le journal mysql et les événements RDS dans AWS Management Console.
- Amélioration du traitement du journal binaire (binlog) pour réduire le temps de récupération sur incident et la latence de temps de validation lorsque de très grandes transactions sont incluses.
- Aurora redimensionne dynamiquement l'espace de stockage de votre cluster. Avec le redimensionnement dynamique, l'espace de stockage de votre cluster de base de données Aurora diminue automatiquement lorsque vous supprimez des données du cluster de base de données. Pour plus d'informations, consultez Dimensionnement du stockage dans le Guide de l'utilisateur Amazon Aurora.

  > **Note**
  >
  > La fonctionnalité de redimensionnement dynamique est déployée par étapes dans les AWS régions où Aurora est disponible. Selon la région où se trouve votre cluster, il se peut que cette fonction ne soit pas encore disponible. Pour plus d'informations, veuillez consulter l'annonce des nouveautés.

Correctifs à priorité élevée :

- CVE-2019-2911
- CVE-2019-2537
- CVE-2018-2787
- CVE-2018-2784

- CVE-2018-2645
- CVE-2018-2640

Améliorations de la disponibilité :

- Correction d'un problème lié au gestionnaire de verrous, où une condition de concurrence pouvait provoquer le partage d'un verrou par deux transactions, provoquant le redémarrage de la base de données.
- Correction d'un problème lié à la gestion de la mémoire de verrouillage des transactions avec des transactions d'écriture longues entraînant un redémarrage de la base de données.
- Correction d'une condition de concurrence dans le gestionnaire de verrous qui entraînait le redémarrage de la base de données ou le basculement pendant la restauration d'une transaction.
- Correction d'un problème lors de la mise à niveau de la version 5.6 vers la version 5.7 lors d'une modification `innodb_file_format` sur une table sur laquelle Fast DLL était activée.
- Correction de plusieurs problèmes dans lesquels le moteur pouvait redémarrer pendant l'application de correctifs sans temps d'arrêt lors de la recherche d'un point de repos dans l'activité de la base de données pour l'application de correctifs.
- Correction d'un problème lié à la récupération DDL qui avait un impact sur le redémarrage de l'instance de base de données lors de la récupération d'une opération `DROP TRIGGER` interrompue.
- Correction d'un bogue qui pouvait entraîner l'indisponibilité de la base de données en cas de plantage lors de l'exécution de certaines opérations de partitionnement. Plus précisément, une opération `ALTER TABLE` interrompue qui modifie le type de partitionnement ou le nombre de partitions dans une table.
- Correction de la valeur par défaut de `table_open_cache` sur les instances 16XL et 24XL qui pouvait provoquer des basculements répétés et une utilisation élevée du processeur sur les classes d'instances volumineuses (R4/R5-16XL, R5-12XL, R5-24XL). Cela a eu un impact sur 1.21.x et 1.22.x.

Bases de données globales :

- Renseignez les données manquantes dans la `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` vue MySQL sur les AWS régions principales et secondaires d'une base de données globale Aurora.

- Correction d'échecs de requête inattendus qui pouvaient se produire dans une région secondaire de base de données globale en raison d'un nettoyage mémoire des enregistrements UNDO dans la région principale, après des problèmes de connectivité réseau temporaires entre les régions principale et secondaire.

Requête parallèle :

- Correction d'un problème dans lequel une requête parallèle pouvait provoquer un retour de résultat vide dans une requête longue.
- Correction d'un problème dans lequel une requête sur une petite table du réplica en lecture Aurora pouvait prendre plus d'une seconde.
- Correction d'un problème qui pouvait provoquer un redémarrage lorsqu'une requête parallèle et une instruction DML s'exécutaient simultanément sous une charge de travail lourde.

Améliorations générales :

- Correction d'un problème où les requêtes utilisant l'index spatial pouvaient renvoyer des résultats partiels si l'index spatial était créé sur des tables avec des valeurs spatiales importantes déjà existantes.
- Augmentation de la longueur maximale autorisée pour les variables du système d'audit `server_audit_incl_users` et `server_audit_excl_users` de 1024 octets à 2000 octets.
- Correction d'un problème dans lequel un réplica de journal binaire connecté à un principal de journal binaire Aurora MySQL pouvait afficher des données incomplètes lorsque le principal de journal binaire Aurora MySQL chargeait les données depuis S3 sous `statement binlog_format`.
- Respectez le comportement de la communauté pour mapper `mixed` binlog_format à `row` au lieu de `statement` pour le chargement des données.
- Correction d'un problème provoquant l'arrêt de la réplication de journal binaire lorsque l'utilisateur ferme la connexion et que la session utilise des tables temporaires.
- Temps de réponse amélioré d'une requête impliquant des tables temporaires MyISAM.
- Correction d'un problème d'autorisation lorsque le travail de journal binaire exécute une fonction lambda native.
- Correction d'un problème sur les réplicas en lecture Aurora lorsque vous tentez d'interroger ou de faire pivoter le journal lent ou le journal général.

- Correction d'un problème qui interrompait la réplication logique lorsque le paramètre `binlog_checksum` était défini sur des valeurs différentes sur l'élément principal et le réplica.
- Correction d'un problème dans lequel le réplica en lecture pouvait voir temporairement les résultats partiels d'une transaction récemment validée sur le rédacteur.
- Incluez les informations de transaction de la transaction annulée dans `show engine innodb status` lorsqu'un blocage est résolu.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Les événements Binlog avec `ALTER TABLE ADD COLUMN ALGORITHM=QUICK` seront réécrits en tant que `ALGORITHM=DEFAULT` de manière à être compatibles avec l'édition de la communauté.
- BOGUE #22350047 : SI LE CLIENT EST TUÉ APRÈS LA RESTAURATION AU POINT DE SAUVEGARDE PRÉCÉDENT STMTS VALIDÉ
- Bogue #29915479 : L'EXÉCUTION DE COM_REGISTER_SLAVE SANS COM_BINLOG_DUMP PEUT GÉNÉRER L'ARRÊT DU SERVEUR
- Bogue #30441969 : BOGUE #29723340 : INCIDENT MYSQL SERVER APRÈS REQUÊTE SQL AVEC ?AST DE DONNÉES
- Bogue #30628268 : INCIDENT DE MÉMOIRE INSUFFISANTE
- Bogue #27081349 : COMPORTEMENT INATTENDU LORS D'UNE SUPPRESSION AVEC FONCTION SPATIALE
- Bogue #27230859 : COMPORTEMENT INATTENDU LORS DE LA GESTION D'UN POLYGONE NON VALIDE
- Bogue #27081349 : COMPORTEMENT INATTENDU LORS DE LA SUPPRESSION AVEC FONCTION SPATIALE
- Bogue #26935001 : AUTO_INCREMENT DE MODIFICATION DE TABLE TENTE DE LIRE L'INDEX À PARTIR D'UN ESPACE DE TABLE SUPPRIMÉ
- Bogue #29770705 : INCIDENT SERVEUR PENDANT L'EXÉCUTION DE SELECT AVEC CLAUSE WHERE SPÉCIFIQUE
- Bogue #27659490 : SELECT AVEC UTILISATION DE PLAGE DYNAMIQUE ET DE FUSION D'INDEX NÉCESSITE TROP DE MÉMOIRE (MÉMOIRE INSUFFISANTE)
- Bogue #24786290 : INTERRUPTION DE LA RÉPLICATION LORSQUE LE BOGUE #74145 SE PRODUIT SUR LE PRINCIPAL

- Bogue #27703912 : UTILISATION DE MÉMOIRE EXCESSIVE AVEC NOMBREUSES PRÉPARATIONS
- Bug #20527363 : CRASH DE LA TABLE TEMPORAIRE TRUNCATE : ! TF2DICT_ _FLAG_IS_SET (TABLE, DICT_ _TEMPORAIRE) TF2
- Bogue #23103937 : PS_TRUNCATE_ALL_TABLES() NE FONCTIONNE PAS EN MODE SUPER_READ_ONLY
- Bogue #25053286 : L'UTILISATION DE VUE AVEC CONDITION DANS LA PROCÉDURE GÉNÈRE UN COMPORTEMENT INCORRECT (corrigé dans 5.6.36)
- Bogue #25586773 : COMPORTEMENT INCORRECT POUR SÉLECTION DE TABLE DE CRÉATION DANS UNE BOUCLE DANS SP (corrigé dans 5.6.39)
- Bogue #27407480 : AUTOMATIC_SP_PRIVILEGES NÉCESSITE LES PRIVILÈGES INSERT POUR LA TABLE MYSQL.USER
- Bogue #26997096 : la valeur de `relay_log_space` n'est pas mise à jour de manière synchronisée, de sorte qu'elle est parfois beaucoup plus élevée que l'espace disque réel utilisé par les journaux relais.
- Bogue #15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY NE FONCTIONNE PAS COMME PRÉVU
- Bogue backport bogue SSL #17087862, bogue #20551271
- Bogue #16894092 : RÉGRESSION DE PERFORMANCE DANS 5.6.6+ POUR INSERT INTO ... SELECT ... FROM (corrigé dans 5.6.15).
- Porter un correctif de bogue lié à `SLAVE_TYPE_CONVERSIONS`.

# Mises à jour du moteur de base de données Aurora MySQL du 03/06/2021 (version 1.22.5) (obsolète)

Version : 1.22.5

Aurora MySQL 1.22.5 est disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une version plus ancienne d'Aurora MySQL, spécifiez la version du moteur via la console RDS, l' AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

Améliorations de la disponibilité :

- Résolution d'un problème qui pouvait entraîner l'arrêt de la base de données, puis un redémarrage ou un basculement en raison d'un conflit de simultanéité entre les threads de nettoyage internes.
- Résolution d'un problème qui pouvait entraîner l'indisponibilité du cluster si la base de données redémarrait tout en conservant les transactions XA à l'état préparé, puis redémarrait à nouveau avant que ces transactions ne soient validées ou annulées. Avant ce correctif, vous pouviez résoudre le problème en restaurant le cluster à un moment donné avant le premier redémarrage.
- Résolution d'un problème qui pouvait entraîner le blocage de la purge InnoDB si la base de données redémarre pendant le traitement d'une instruction DDL. Par conséquent, la longueur de la liste de l'historique InnoDB augmenterait et le volume de stockage du cluster continuerait de croître jusqu'à ce qu'il soit rempli, rendant ainsi la base de données indisponible. Avant ce correctif, vous pouviez atténuer le problème en redémarrant à nouveau la base de données pour débloquer la purge.

## Mises à jour du moteur de base de données Aurora MySQL du 04/02/2021 (version 1.22.4) (obsolète)

Version : 1.22.4

Aurora MySQL 1.22.4 est disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- CVE-2020-14867
- CVE-2020-14812
- CVE-2020-14793
- CVE-2020-14769
- CVE-2020-14765

- CVE-2020-14672
- CVE-2020-1971

Améliorations de la disponibilité :

- Correction d'un problème susceptible de déclencher un redémarrage ou un basculement de la base de données au cours d'une commande `kill session`. Si vous rencontrez ce problème, contactez le AWS support pour activer ce correctif sur votre instance.
- Amélioration de la journalisation binaire pour réduire le temps de récupération sur incident et la latence de validation lorsque de grandes transactions sont incluses.
- Correction d'un problème qui provoquait l'arrêt d'un réplica de journal binaire avec une erreur `HA_ERR_KEY_NOT_FOUND`.

# Mises à jour du moteur de base de données Aurora MySQL du 19/11/2020 (version 1.22.3) (obsolète)

Version : 1.22.3

Aurora MySQL 1.22.3 est disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, spécifiez la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs de sécurité :

Correctifs et autres améliorations visant à peaufiner la gestion dans un environnement géré. Correctifs CVE supplémentaires ci-dessous :

- CVE-2020-14559
- CVE-2020-14539
- CVE-2020-2579
- CVE-2020-2812
- CVE-2020-2780
- CVE-2020-2763

Modifications incompatibles :

Cette version introduit une modification d'autorisation qui affecte le comportement de la commande `mysqldump`. Les utilisateurs doivent disposer du privilège `PROCESS` pour accéder à la table `INFORMATION_SCHEMA.FILES`. Pour exécuter la commande `mysqldump` sans aucune modification, accordez le privilège `PROCESS` à l'utilisateur de base de données auquel la commande `mysqldump` se connecte. Vous pouvez également exécuter la commande `mysqldump` avec l'option `--no-tablespaces`. Avec cette option, la sortie `mysqldump` n'inclut aucune instruction `CREATE LOGFILE GROUP` ou `CREATE TABLESPACE`. Dans ce cas, la commande `mysqldump` n'accède pas à la table `INFORMATION_SCHEMA.FILES` et vous n'avez pas besoin d'accorder l'autorisation `PROCESS`.

Améliorations de la disponibilité :

- Correction de problèmes pouvant entraîner le redémarrage du serveur lors de la restauration d'une instruction DDL qui n'a pas été validée.
- Correction des conditions de concurrence dans le gestionnaire de verrous pouvant provoquer le redémarrage du serveur.

- Correction d'un problème pouvant entraîner le redémarrage du serveur par l'agent de surveillance lors de la récupération d'une transaction volumineuse.

Améliorations générales :

- Comportement modifié pour mapper `MIXED` `binlog_format` sur `ROW` au lieu de `STATEMENT` lors de l'exécution de `LOAD DATA FROM INFILE | S3`.
- Correction d'un problème dans lequel un réplica de journal binaire connecté à un principal de journal binaire Aurora MySQL pouvait afficher des données incomplètes lorsque le principal exécutait `LOAD DATA FROM S3` et `binlog_format` défini sur `STATEMENT`.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue n° 26654685 : Un ID d'index corrompu rencontré lors d'une vérification de clé étrangère déclenchait une assertion.
- Bogue n° 15831300 : Par défaut, lors de la promotion de nombres entiers d'un type plus petit sur le maître à un type plus grand sur l'esclave (par exemple, d'une colonne SMALLINT sur le maître à une colonne BIGINT sur l'esclave), les valeurs promues sont traitées comme si elles étaient signées. Dans de tels cas, il est possible de modifier ou de remplacer ce comportement à l'aide de `ALL_SIGNED`, de `ALL_UNSIGNED` ou des deux dans l'ensemble des valeurs spécifiées pour la variable système serveur slave_type_conversions. Pour en savoir plus, consultez la section Row-based replication: attribute promotion and demotion, ainsi que la description de la variable.
- Bogue n° 17449901 : Avec `foreign_key_checks=0`, InnoDB permettait de supprimer un index requis par une contrainte de clé étrangère, plaçant la table dans une incohérence et provoquant l'échec de la vérification de la clé étrangère lors du chargement de la table. InnoDB empêche désormais de supprimer un index requis par une contrainte de clé étrangère, même avec foreign_key_checks=0. La contrainte de clé étrangère doit être supprimée avant de supprimer l'index de clé étrangère.
- BOGUE n° 20768847 : Une opération ALTER TABLE ... Opération DROP INDEX sur une table avec des dépendances de clé étrangère déclenchait une assertion.

# Mises à jour du moteur de base de données Aurora MySQL du 05/03/2020 (version 1.22.2) (obsolète)

Version : 1.22.2

Aurora MySQL 1.22.2 est en disponibilité générale. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sa disponibilité fera l'objet d'une annonce distincte.
> Cette version est désignée comme version de support à long terme (LTS). Pour plus d'informations, consultez Versions Long-Term Support (LTS) d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs à priorité élevée :

- Correction d'un problème d'échecs intermittents de connexion après la rotation du certificat.
- Correction d'un problème qui entraînait un clonage plus long sur certains clusters de bases de données avec des charges d'écriture élevées.

- Correction d'un problème qui interrompait la réplication logique lorsque le paramètre `binlog_checksum` était défini sur des valeurs différentes sur l'élément principal et le réplica.
- Correction d'un problème de rotation des journaux lents et des journaux généraux sur les réplicas en lecture.
- Correction d'un problème avec le comportement du niveau d'isolation Read Committed ANSI.

# Mises à jour du moteur de base de données Aurora MySQL du 23/12/2019 (version 1.22.1) (obsolète)

Version : 1.22.1

Aurora MySQL 1.22.1 est en général disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de l'API AWS CLI ou de l'API RDS. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 1.* existants vers Aurora MySQL 1.22.1.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Chine (Ningxia) [cn-northwest-1], Asie-Pacifique (Hong Kong) [ap-east-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

Correctifs critiques :

- Correction de problèmes qui empêchaient la récupération du moteur impliquant des verrous de table et des tables temporaires.
- Amélioration de la stabilité du journal binaire lorsque des tables temporaires sont utilisées.

Correctifs à priorité élevée :

- Correction d'une fuite de mémoire lente dans le sous-système de suivi et de journalisation de base de données spécifique à Aurora qui réduisait la mémoire libérable.

## Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 1.22.0) (obsolète)

Version : 1.22.0

Aurora MySQL 1.22.0 est en général disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de l'API AWS CLI ou de l'API RDS. Vous

avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 1.* existants vers Aurora MySQL 1.22.0.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Chine (Ningxia) [cn-northwest-1], Asie-Pacifique (Hong Kong) [ap-east-1], Moyen-Orient (Bahreïn) [me-south-1] et Amérique du Sud (São Paulo) [sa-east-1] [1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Nouvelles fonctions :

- Les clusters Aurora MySQL prennent désormais en charge les types d'instance db.r5.8xlarge, db.r5.16xlarge et db.r5.24xlarge.
- Nouvelles améliorations apportées aux journaux binaires (binlog) pour une meilleure latence de validation lorsque de très larges transactions sont impliquées.
- Aurora MySQL possède désormais un mécanisme pour réduire la fenêtre de temps pendant laquelle les événements d'une transaction volumineuse sont écrits sur les journaux binaires à la validation. Cette solution empêche les interminables récupérations hors connexion lorsque les arrêts de base de données se produisent pendant cette fenêtre temporelle. Cette fonctionnalité corrige aussi le problème où une transaction volumineuse bloque les petites transactions sur la validation des journaux binaires. Cette fonctionnalité est désactivée par défaut et peut être activée

par l'équipe du service si nécessaire pour la charge de travail. Lorsqu'elle est activée, elle est déclenchée lorsqu'une taille de transaction est > 500 Mo.

- Ajout de la prise en charge pour le niveau d'isolation ANSI `READ COMMITTED` sur les réplicas en lecture. Ce niveau d'isolation permet aux requêtes de longue durée sur le réplica en lecture de s'exécuter sans impacter le haut débit des écritures sur le nœud d'écriture. Pour plus d'informations, veuillez consulter Niveaux d'isolation Aurora MySQL.
- Les bases de données mondiales permettent désormais d'ajouter des régions de réplication secondaires en lecture seule pour les clusters de bases de données déployés dans ces AWS régions : régions : USA Est (Virginie du Nord) [us-east-1], USA Est (Ohio) [us-east-2], USA Ouest (Californie du Nord) [us-west-1], USA Ouest (Oregon) [us-west-2], Europe (Irlande) [eu-west-2] eu-west-1], Europe (Londres) [eu-west-2], Europe (Paris) [eu-west-3], Asie-Pacifique (Tokyo) [ap-northeast-1], Asie-Pacifique (Séoul) [ap-northeast-2], Asie-Pacifique (Singapour) [ap-northeast-2] ap-southeast-1], Asie-Pacifique (Sydney) [ap-southeast-2], Canada (centre) [ca-central-1], Europe (Francfort) [ eu-central-1] et Asie-Pacifique (Mumbai) [ap-south-1].
- La fonctionnalité de conflits entre lignes critiques est désormais disponible et ne nécessite pas que le mode lab Aurora soit activé (ON). Cette fonction améliore nettement le débit pour les charges où de nombreuses transactions sont en conflit pour les lignes d'une même page.
- Cette version possède des fichiers de fuseau horaire mis à jour pour prendre en charge la dernière mise à jour du fuseau du Brésil pour les nouveaux clusters.

Correctifs critiques :

- CVE-2019-2922
- CVE-2019-2923
- CVE-2019-2924
- CVE-2019-2910

Correctifs à priorité élevée :

- 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
- Correction du problème du composant de récupération de DLL qui entraînait un arrêt prolongé de la base de données. Les clusters devenus indisponibles après l'exécution d'une requête `TRUNCATE TABLE` sur une table avec une colonne `AUTO_INCREMENT` doivent être mis à jour.
- Correction du problème du composant de récupération de DLL qui entraînait un arrêt prolongé de la base de données. Les clusters devenus indisponibles après l'exécution d'une requête `DROP TABLE` sur plusieurs tables en parallèle doivent être mis à jour.

Correctifs de stabilité générale :

- Correction d'un problème qui provoquait le redémarrage de réplicas en lecture pendant une transaction de longue durée. Les clients confrontés à des démarrages de réplicas coïncidant avec une suppression accélérée dans la mémoire libérable doivent envisager de mettre cette version à niveau.
- Correction d'un problème signalant à tort une erreur `ERROR 1836` lorsqu'une requête imbriquée est exécutée sur une table temporaire sur le réplica en lecture.
- Résolution d'une erreur d'annulation de requête sur les instances de lecteur Aurora pendant l'exécution d'une charge de travail intense en écriture sur l'instance d'écriture Aurora.
- Correction du problème qui entraîne le redémarrage d'une base de données configurée comme « journal binaire principal (binlog) » tandis qu'une charge massive de travail en écriture est en cours d'exécution.
- Correction du problème d'indisponibilité prolongée lors du redémarrage du moteur. La correction concerne un problème d'initialisation du pool de mémoires tampon. Ce problème se produit rarement, mais peut éventuellement impacter toute version prise en charge.
- Correction d'un problème qui générait des données non cohérentes dans la table `information_schema.replica_host_status`.
- Correction d'une condition de course entre la requête parallèle et les chemins d'exécution standard qui entraînait le redémarrage intermittent des nœuds en lecture.
- Amélioration de la stabilité de la base de données lorsque le nombre de connexions clients dépasse la valeur du paramètre `max_connections`.

- Amélioration de la stabilité des instances en lecture grâce au blocage des DLL non prises en charge et des requêtes `LOAD FROM S3`.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue #16346241 - ARRÊT DU SERVEUR DANS ITEM_PARAM::QUERY_VAL_STR
- Bogue #17733850 - ARRÊT DE NAME_CONST() DANS ITEM_NAME_CONST::ITEM_NAME_CONST()
- Bogue #20989615: INNODB AUTO_INCREMENT GÉNÈRE DEUX FOIS LA MÊME VALEUR
- Bogue #20181776 - LE CONTRÔLE D'ACCÈS NE CORRESPOND PAS À L'HÔTE LE PLUS SPÉCIFIQUE QUAND IL CONTIENT DES CARACTÈRES GÉNÉRIQUES
- Bogue #27326796 - ARRÊT DE MYSQL AVEC ÉCHEC DE L'ASSERTION INNODB DANS LE FICHIER PARS0PARS.CC
- Bug #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL

## Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 1.21.0) (obsolète)

Version : 1.21.0

Aurora MySQL 1.21.0 est en général disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

Versions Aurora MySQL actuellement prises en charge : 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.* et 2.07.*. Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de l'API AWS CLI ou de l'API RDS. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 1.* existants vers Aurora MySQL 1.21.0.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Chine (Ningxia) [cn-northwest-1], Asie-Pacifique (Hong Kong) [ap-east-1], Europe (Stockholm) [eu-

north-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs critiques :

- CVE-2018-0734
- CVE-2019-2534
- CVE-2018-2612
- CVE-2017-3599
- CVE-2018-2562
- CVE-2017-3329
- CVE-2018-2696
- CVE-2015-4737

Correctifs à priorité élevée :

- Il est vivement recommandé aux clients dont la taille de la base de données est proche de 64 tébioctets (TiO) de procéder à une mise à niveau vers cette version pour éviter les arrêts dus aux bogues de stabilité affectant les volumes proches de la limite de stockage Aurora.

Correctifs de stabilité générale :

- Résolution d'une erreur d'annulation de requête sur les instances de lecteur Aurora pendant l'exécution d'une charge de travail intense en écriture sur l'instance d'écriture Aurora.
- Résolution d'un problème sur les instances de lecteur Aurora qui ont réduit la mémoire disponible pendant des transactions longues alors qu'il y a un trafic important d'engagement des transactions sur l'instance d'écriture.
- La valeur du paramètre `aurora_disable_hash_join` est maintenant conservée après le redémarrage de la base de données ou le remplacement de l'hôte.
- Résolution d'un problème lié au cache de recherche de la totalité du texte en raison duquel l'instance Aurora a manqué de mémoire. Les clients qui utilisent la recherche en texte intégral doivent procéder à une mise à niveau.
- Amélioration de la stabilité de la base de données lorsque la fonction de jointure de hachage est activée et que l'instance dispose de peu de mémoire. Les clients qui utilisent la jointure par hachage doivent procéder à une mise à niveau.
- Résolution d'un problème dans le cache de requête selon lequel l'erreur « Trop de connexions » pouvait entraîner un redémarrage.
- Résolution du calcul de mémoire disponible sur les instances T2 de façon à inclure un espace d'échange de mémoire pour empêcher les redémarrages superflus.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bug #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) DANS __MEMMOVE_ _BACK FROM STRING : :COPY SSSE3
- Bogue n° 17059925 : pour les instructions UNION, la valeur examinée par les lignes a été calculée de façon incorrecte. Cela a donné des valeurs trop importantes pour la colonne `ROWS_EXAMINED` des tables de l'instruction Schéma des performances (par exemple, events_statements_current).
- Bogue n° 11827369 : certaines requêtes avec des sous-requêtes imbriquées `SELECT ... FROM DUAL` ont généré une assertion.
- Bogue n° 16311231 : des résultats incorrects ont été renvoyés lorsqu'une requête contenait une sous-requête dans une clause `IN` qui contenait une opération XOR dans la clause `WHERE`.

# Mises à jour du moteur de base de données Aurora MySQL du 05/03/2020 (version 1.20.1) (obsolète)

Version : 1.20.1

Aurora MySQL 1.20.1 est en disponibilité générale. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

Versions Aurora MySQL actuellement prises en charge : 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.* et 2.07.*. Vous pouvez restaurer l'instantané d'une base de données Aurora MySQL 1.* à la version Aurora MySQL 1.20.1.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs à priorité élevée :

- Correction d'un problème d'échecs intermittents de connexion après la rotation du certificat.
- Correction d'un problème lié à la simultanéité de la fermeture des connexions qui entraînerait un basculement sur incident en cas de charge de travail importante.

Correctifs de stabilité générale :

- Correction d'un incident lors de l'exécution d'une requête complexe impliquant une agrégation et des jointures de plusieurs tables utilisant en interne des tables intermédiaires.

# Mises à jour du moteur de base de données Aurora MySQL du 11/11/2019 (version 1.20.0) (obsolète)

Version : 1.20.0

Aurora MySQL 1.20.0 est en général disponible. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

Versions Aurora MySQL actuellement prises en charge : 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 1.20.*, 2.01.*, 2.02.*, 2.03.* et 2.04.*. Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur par le biais de l'API AWS Management Console, de l'API AWS CLI ou de l'API RDS. Vous avez la possibilité de mettre à niveau les clusters de bases de données Aurora MySQL 1.* existants vers 1.19.5 ou Aurora MySQL 1.20.0.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Chine (Ningxia) [cn-northwest-1], Asie-Pacifique (Hong Kong) [ap-east-1], Europe (Stockholm) [eu-north-1] et Moyen-Orient (Bahreïn) [me-south-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

Correctifs critiques :

- CVE-2018-0734
- CVE-2019-2534
- CVE-2018-2612
- CVE-2017-3599
- CVE-2018-2562
- CVE-2017-3329
- CVE-2018-2696
- CVE-2015-4737

Correctifs à priorité élevée :

- Il est vivement recommandé aux clients dont la taille de la base de données est proche de 64 tébioctets (TiO) de procéder à une mise à niveau vers cette version pour éviter les arrêts dus aux bogues de stabilité affectant les volumes proches de la limite de stockage Aurora.

Correctifs de stabilité générale :

- Résolution d'une erreur d'annulation de requête sur les instances de lecteur Aurora pendant l'exécution d'une charge de travail intense en écriture sur l'instance d'écriture Aurora.
- Résolution d'un problème sur les instances de lecteur Aurora qui ont réduit la mémoire disponible pendant des transactions longues alors qu'il y a un trafic important d'engagement des transactions sur l'instance d'écriture.
- La valeur du paramètre `aurora_disable_hash_join` est maintenant conservée après le redémarrage de la base de données ou le remplacement de l'hôte.
- Résolution d'un problème lié au cache de recherche de la totalité du texte en raison duquel l'instance Aurora a manqué de mémoire. Les clients qui utilisent la recherche en texte intégral doivent procéder à une mise à niveau.
- Amélioration de la stabilité de la base de données lorsque la fonction de jointure de hachage est activée et que l'instance dispose de peu de mémoire. Les clients qui utilisent la jointure par hachage doivent procéder à une mise à niveau.
- Résolution d'un problème dans le cache de requête selon lequel l'erreur « Trop de connexions » pouvait entraîner un redémarrage.
- Résolution du calcul de mémoire disponible sur les instances T2 de façon à inclure un espace d'échange de mémoire pour empêcher les redémarrages superflus.

## Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bug #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) DANS __MEMMOVE_ _BACK FROM STRING : :COPY SSSE3
- Bogue n° 17059925 : pour les instructions UNION, la valeur examinée par les lignes a été calculée de façon incorrecte. Cela a donné des valeurs trop importantes pour la colonne `ROWS_EXAMINED` des tables de l'instruction Schéma des performances (par exemple, events_statements_current).
- Bogue n° 11827369 : certaines requêtes avec des sous-requêtes imbriquées `SELECT ... FROM DUAL` ont généré une assertion.
- Bogue n° 16311231 : des résultats incorrects ont été renvoyés lorsqu'une requête contenait une sous-requête dans une clause `IN` qui contenait une opération XOR dans la clause `WHERE`.

# Mises à jour du moteur de base de données Aurora MySQL du 05/03/2020 (version 1.19.6) (obsolète)

Version : 1.19.6

Aurora MySQL 1.19.6 est en disponibilité générale. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous pouvez restaurer l'instantané d'une base de données Aurora MySQL 1.* à la version Aurora MySQL 1.19.6.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

Correctifs à priorité élevée :

- Correction d'un problème d'échecs intermittents de connexion après la rotation du certificat.

## Mises à jour du moteur de base de données Aurora MySQL du 19/09/2019 (version 1.19.5) (obsolète)

Version : 1.19.5

Aurora MySQL 1.19.5 est disponible pour le grand public. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

L'obsolescence de cette version du moteur est prévue pour le 28 février 2023. Pour plus d'informations, consultez Préparation à la fin de vie d'Amazon Aurora Édition compatible avec MySQL version 1.

Les versions d'Aurora MySQL actuellement prises en charge sont les suivantes : 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* et 3.02.*.

Vous avez la possibilité de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.19.5. Vous pouvez restaurer les instantanés d'Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.1 et 1.19.2 dans Aurora MySQL 1.19.5.

Pour utiliser une ancienne version d'Aurora MySQL, vous pouvez créer de nouveaux clusters de bases de données en spécifiant la version du moteur par le biais de l'API AWS Management Console, de AWS CLI, ou de l'API RDS.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les AWS régions suivantes : Europe (Londres) [eu-west-2] AWS GovCloud , (US-Est) [-1], (US-Ouest) us-gov-east [-1] AWS GovCloud , Chine (Ningxia) us-gov-west [cn-northwest-1] et Asie-Pacifique (Hong Kong) [ap-east-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Résolution d'un problème sur les instances de lecteur Aurora qui ont réduit la mémoire disponible pendant des transactions longues alors qu'il y a un trafic important d'engagement des transactions sur l'instance d'écriture.
- Résolution d'une erreur d'annulation de requête sur les instances de lecteur Aurora pendant l'exécution d'une charge de travail intense en écriture sur l'instance d'écriture Aurora.
- La valeur du paramètre `aurora_disable_hash_join` est maintenant conservée après le redémarrage de la base de données ou le remplacement de l'hôte.
- Résolution d'un problème lié au cache de recherche de la totalité du texte en raison duquel l'instance Aurora a manqué de mémoire.
- Amélioration de la stabilité de la base de données lorsque la taille du volume est proche de la limite des 64 tébioctets (Tio) en réservant 160 Go d'espace pour que le flux de travail soit réalisé sans basculement.
- Amélioration de la stabilité de la base de données lorsque la fonction de jointure de hachage est activée et que l'instance dispose de peu de mémoire.

- Résolution du calcul de mémoire disponible de façon à inclure un espace d'échange de mémoire sur les instances T2 ayant entraîné un redémarrage prématuré.
- Résolution d'un problème dans le cache de requête selon lequel l'erreur « Trop de connexions » pouvait entraîner un redémarrage.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

- CVE-2018-2696
- CVE-2015-4737
- Bug #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) DANS __MEMMOVE_ _BACK FROM STRING : :COPY SSSE3
- Bogue n° 17059925 : pour les instructions UNION, la valeur examinée par les lignes a été calculée de façon incorrecte. Cela a donné des valeurs trop importantes pour la colonne `ROWS_EXAMINED` des tables de l'instruction Schéma des performances (par exemple, events_statements_current).
- Bogue n° 11827369 : certaines requêtes avec des sous-requêtes imbriquées `SELECT ... FROM DUAL` ont généré une assertion.
- Bogue n° 16311231 : des résultats incorrects ont été renvoyés lorsqu'une requête contenait une sous-requête dans une clause `IN` qui contenait une opération XOR dans la clause `WHERE`.

## Mises à jour du moteur de base de données Aurora MySQL du 05/06/2019 (version 1.19.2) (obsolète)

Version : 1.19.2

Aurora MySQL 1.19.2 est généralement disponible. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, peuvent être créés avec 1.17.8, 1.19.0, 1.19.1 ou 1.19.2. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.19.2. Pour utiliser une version antérieure, vous pouvez créer des clusters de bases de données dans Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 ou Aurora MySQL 1.18. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (US-West) [us-gov-west-1], Europe (Stockholm) [eu-north-1], Chine (Ningxia) [cn-northwest-1] et Asie-Pacifique (Hong Kong) [ap-east-1]. AWS Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

- Résolution d'un problème qui pouvait entraîner des échecs lors du chargement de données vers Aurora à partir de Amazon S3.
- Résolution d'un problème qui pouvait entraîner des échecs lors du chargement de données à partir d'Aurora vers Amazon S3.
- Résolution d'un problème qui créait des sessions zombie à l'état Supprimé.
- Résolution d'un problème qui interrompait les connexions lors du traitement d'une erreur de gestion du protocole réseau.
- Résolution d'un problème qui pouvait entraîner un blocage lors du traitement de tables partitionnées.
- Résolution d'un problème lié à la réplication des journaux binaires de la création des déclencheurs.

## Mises à jour du moteur de base de données Aurora MySQL du 09/05/2019 (version 1.19.1) (obsolète)

Version : 1.19.1

Aurora MySQL 1.19.1 est en disponibilité générale. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, peuvent être créés avec 1.17.8, 1.19.0 ou 1.19.1. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.19.1. Pour utiliser une version antérieure, vous pouvez créer des clusters de bases de données dans Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 ou Aurora MySQL 1.18. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'un bogue dans la réplication des journaux binaires pouvant entraîner un problème sur les instances Aurora configurées en tant que travailleur des journaux binaires.
- Correction d'une erreur survenant lors de la gestion de certains types de commandes `ALTER TABLE`.
- Correction d'une erreur d'interruption des connexions en raison d'une erreur de gestion du protocole réseau.

# Mises à jour du moteur de base de données Aurora MySQL du 07/02/2019 (version 1.19.0) (obsolète)

Version : 1.19.0

Aurora MySQL 1.19.0 est en disponibilité générale. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, peuvent être créés avec 1.17.8 ou 1.19.0. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.19.0. Pour utiliser une version antérieure, vous pouvez créer des clusters de bases de données dans Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 ou Aurora MySQL 1.18.0. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

> **Note**
>
> La procédure de mise à niveau du cluster de bases de données a changé. Pour plus d'informations, consultez Mise à niveau de la version mineure ou du niveau de correctif d'un cluster de bases de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Fonctionnalités

- Sélecteur de version Aurora - Depuis la version Aurora MySQL 1.19.0, vous avez le choix entre plusieurs versions Aurora compatibles avec MySQL 5.6 dans la console Amazon RDS. Pour plus d'informations, consultez Vérification ou spécification de versions du moteur Aurora MySQL via AWS dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Résolution d'un problème de stabilité lié à la requête `CHECK TABLE` sur un réplica Aurora.
- Lancement d'une nouvelle variable utilisateur global `aurora_disable_hash_join` pour désactiver la jointure par hachage.
- Résolution d'un problème de stabilité lors de la génération de la ligne de sortie pendant la jointure par hachage de plusieurs tables.
- Résolution d'un problème entraînant un résultat incorrect en raison d'un changement de plan pendant la vérification de l'applicabilité d'une jointure de hachage.
- L'application de correctifs sans temps d'arrêt est prise en charge avec les transactions de longue durée. Cette amélioration prendra effet lors de la mise à niveau de la version 1.19 à une version supérieure.
- L'application de correctifs sans temps d'arrêt est désormais prise en charge lorsque la journalisation binaire est activée. Cette amélioration prendra effet lors de la mise à niveau de la version 1.19 à une version supérieure.
- Résolution d'un problème qui entraînait un pic d'utilisation de l'UC sur le réplica Aurora non associé à la charge de travail.
- Correction d'une condition de concurrence dans le gestionnaire de verrous qui entraînait le redémarrage de la base de données.
- Correction d'une condition de concurrence dans le composant du gestionnaire de verrous pour améliorer la stabilité des instances Aurora.
- Amélioration de la stabilité du détecteur d'interblocage dans le composant du gestionnaire de verrous.
- `INSERT`L'opération sur une table est interdite si InnoDB détecte que l'index est corrompu.
- Résolution d'un problème de stabilité dans Fast DDL.
- Amélioration de la stabilité Aurora par réduction de la consommation de la mémoire dans le traitement par lot des analyses pour une sous-requête à une seule ligne.
- Résolution d'un problème de stabilité survenant après la suppression d'une clé étrangère alors que la variable système `foreign_key_checks` est définie sur 0.
- Résolution d'un problème de la fonction d'évitement de mémoire insuffisante entraînant le remplacement par erreur des modifications apportées à la valeur `table_definition_cache` par l'utilisateur.
- Résolution de problèmes de stabilité de la fonction d'évitement de mémoire insuffisante.

- Résolution d'un problème qui définissait `query_time` et `lock_time` dans `slow_query_log` en valeurs de garbage.
- Résolution d'un problème de stabilité des requêtes parallèles déclenché par une gestion incorrecte du classement des chaînes en interne.
- Résolution d'un problème de stabilité des requêtes parallèles déclenché par une recherche d'index secondaire.
- Résolution d'un problème de stabilité des requêtes parallèles déclenché par la mise à jour de plusieurs tables.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue n°32917 : DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY
- Bogue n°63144 : CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

## Mises à jour du moteur de base de données Aurora MySQL du 20/09/2018 (version 1.18.0) (obsolète)

Version : 1.18.0

Aurora MySQL 1.18.0 est en général disponible. Tous les nouveaux clusters compatibles avec les requêtes parallèles Aurora MySQL et compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.18.0. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters compatibles avec les requêtes parallèles vers Aurora MySQL 1.18.0. Vous pouvez créer de nouveaux clusters de bases de données dans Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16 ou Aurora MySQL 1.17.6. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.18.0 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

> **Important**
>
> Aurora MySQL 1.18.0 s'applique uniquement aux clusters compatibles avec les requêtes parallèles Aurora. Si vous mettez à niveau un cluster 5.6.10a alloué, la version résultante est

1.17.8. Si vous mettez à niveau un cluster 5.6.10a compatible avec les requêtes parallèles, la version résultante est 1.18.0.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Fonctionnalités

- La fonction de Requête parallèle est disponible avec cette version, pour les nouveaux clusters et les instantanés restaurés. La requête parallèle Aurora MySQL est une optimisation qui parallélise une partie des calculs impliqués dans le traitement I/O des requêtes gourmandes en données. Les tâches qui sont mises en parallèle incluent la récupération des lignes à partir du stockage, l'extraction des valeurs des colonnes et la détermination des lignes répondant aux conditions définies dans la clause `WHERE` et dans les clauses JOIN. Ces tâches de gestion des gros volumes de données sont déléguées à différents nœuds de la couche de stockage distribué Aurora. Dans le jargon, on parle également d'optimisation de base de données par pushdown. Sans la fonction de requête parallèle, chaque requête envoie toutes les données analysées à un même nœud au sein du cluster Aurora MySQL (le nœud principal) et y effectue toutes les tâches de traitement nécessaires.
  - Lorsque la fonction de requête parallèle est activée, le moteur Aurora MySQL détermine automatiquement quand utiliser cette dernière, sans nécessiter de modifications SQL telles que les indicateurs ou les attributs de table.

  Pour plus d'informations, consultez Utilisation des requêtes parallèles pour Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- La fonction d'évitement de mémoire insuffisante surveille la mémoire système et suit la mémoire consommée par les divers composants de la base de données. Lorsque le système manque de mémoire, il effectue une liste d'actions pour libérer de la mémoire à partir des divers composants suivis afin d'essayer d'éviter un dépassement de mémoire de la base de données et donc un redémarrage de cette base de données. Cette fonction d'optimisation est activée par défaut pour les instances t2 et peut être activée pour les autres classes d'instance via un nouveau paramètre d'instance nommé `aurora_oom_response`. Ce paramètre d'instance accepte une chaîne d'actions séparées par des virgules, qu'une instance doit effectuer lorsque sa mémoire est faible. Les actions valides incluent « print », « tune », « decline », « kill_query » et toute combinaison de ces valeurs. Une chaîne vide signifie qu'aucune action ne doit être effectuée et revient en fait

à désactiver la fonction. Notez que les actions par défaut de cette fonction sont « print, tune ». Exemples d'utilisation :

- « print » – imprime uniquement les requêtes exigeant une grande quantité de mémoire.
- « tune » – affine les caches de table interne pour restituer de la mémoire au système.
- « decline » – refuse les nouvelles requêtes une fois que l'instance manque de mémoire.
- « kill_query » – arrête les requêtes dans l'ordre décroissant de leur consommation de mémoire jusqu'à ce que la mémoire de l'instance passe au-dessus du seuil bas. Les instructions en langage de manipulation de données (DDL) ne sont pas abandonnées.
- « print, tune » – effectue les actions décrites pour « print » et « tune ».
- « tune, decline, kill_query » – effectue les actions décrites pour « tune », « decline » et « kill_query ».

Pour plus d'informations sur out-of-memory les conditions de manipulation et d'autres conseils de dépannage, consultez la section Problèmes de mémoire insuffisante d'Amazon Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

# Mises à jour du moteur de base de données Aurora MySQL du 05/03/2020 (version 1.17.9) (obsolète)

Version : 1.17.9

Aurora MySQL 1.17.9 est en disponibilité générale. Les versions 1.x d'Aurora MySQL sont compatibles avec MySQL 5.6 et les versions 2.x d'Aurora MySQL sont compatibles avec MySQL 5.7.

Versions Aurora MySQL actuellement prises en charge : 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.* et 2.07.*. Vous pouvez restaurer l'instantané d'une base de données Aurora MySQL 1.* à la version Aurora MySQL 1.17.9.

Pour créer un cluster avec une ancienne version d'Aurora MySQL, veuillez spécifier la version du moteur via la console RDS, la AWS CLI ou l'API Amazon RDS.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions suivantes : AWS GovCloud (US-East) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

Correctifs à priorité élevée :

- Correction d'un problème d'échecs intermittents de connexion après la rotation du certificat.

## Mises à jour du moteur de base de données Aurora MySQL du 17/01/2019 (version 1.17.8) (obsolète)

Version : 1.17.8

Aurora MySQL 1.17.8 est en général disponible. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.17.8. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.17.8. Pour utiliser une version antérieure, vous pouvez créer des clusters de bases de données dans Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.7. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17.8 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

- Résolution d'un problème de performances qui augmentait l'utilisation de l'UC sur un réplica Aurora après un redémarrage.
- Correction d'un problème de stabilité pour les requêtes `SELECT` qui utilisaient une jonction de hachage.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

- BOGUE 13418638 : CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

## Mises à jour du moteur de base de données Aurora MySQL du 08/10/2018 (version 1.17.7) (obsolète)

Version : 1.17.7

Aurora MySQL 1.17.7 est en général disponible. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.17.7. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.17.7. Pour utiliser une version antérieure, vous pouvez créer des clusters de bases de données dans Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.6. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17.7 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

- La variable d'état InnoDB `innodb_buffer_pool_size` a été rendue visible publiquement pour permettre aux clients de la modifier.
- Correction d'un problème de stabilité sur le cluster Aurora, qui se produisait lors des basculements.
- Amélioration de la disponibilité du cluster en corrigeant un problème de récupération DDL qui se produisait après l'échec d'une opération `TRUNCATE`.
- Correction d'un problème de stabilité lié à la mise à jour de la table `mysql.innodb_table_stats`, déclenché par les opérations DDL.
- Correction des problèmes de stabilité des réplicas Aurora déclenchés lors de l'invalidation du cache de requêtes après une opération DDL.
- Correction d'un problème de stabilité déclenché par un accès à la mémoire non valide pendant l'expulsion régulière du cache de dictionnaire en arrière-plan.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Bogue n° 16208542 : une opération Drop Index sur une colonne de clé étrangère entraîne un problème de table manquante.
- Bogue n° 76349 : fuite de mémoire dans add_derived_key().
- Bogue n° 16862316 : pour les tables partitionnées, les requêtes peuvent renvoyer différents résultats en fonction de l'utilisation ou non de la fusion d'index.
- Bogue n° 17588348 : les requêtes utilisant l'optimisation index_merge (consultez Index Merge Optimization) peuvent renvoyer des résultats non valides lorsqu'elles sont exécutées sur des tables qui ont été partitionnées par HASH.

## Mises à jour du moteur de base de données Aurora MySQL du 06/09/2018 (version 1.17.6) (obsolète)

Version : 1.17.6

Aurora MySQL 1.17.6 est en général disponible. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.17.6. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.17.6. Pour utiliser une version antérieure, vous pouvez créer des clusters de bases de données dans Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.5. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17.6 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'un problème de stabilité sur le lecteur Aurora pour les requêtes `SELECT` pendant que l'enregistreur Aurora exécute des opérations DDL sur la même table.
- Correction d'un problème de stabilité causé par la création et la suppression de journaux DDL pour les tables temporaires utilisant le Heap/Memory moteur.
- Correction d'un problème de stabilité sur le travailleur des journaux binaires lorsque des instructions DDL sont répliquées pendant que la connexion au maître des journaux binaires est instable.
- Correction d'un problème de stabilité rencontré lors de l'écriture dans le journal des requêtes lentes.
- Correction d'un problème lié à la table d'états de réplica qui exposait des informations de retard du lecteur Aurora incorrectes.

### Intégration de correctifs de bogues de l'édition MySQL Community Edition

- Pour une instruction ALTER TABLE qui renommait ou modifiait la valeur par défaut d'une colonne BINARY, l'altération était effectuée en utilisant une copie de la table et non sur place. (Bogues n ° 67141, 14735373, 69580 et 17024290)
- Une jointure externe entre une table standard et une table dérivée qu'elle regroupe implicitement peut entraîner l'arrêt du serveur. (Bogue n° 16177639)

## Mises à jour du moteur de base de données Aurora MySQL du 14/08/2018.(version 1.17.5) (obsolète)

Version : 1.17.5

Aurora MySQL 1.17.5 est en général disponible. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.17.5. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.17.5. Pour utiliser une version antérieure, vous pouvez créer des clusters de bases de données dans Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.4. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17.5 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Résolution d'un problème à cause duquel un enregistreur Aurora peut redémarrer après la correction d'un cluster Aurora à l'aide de la fonction d'application de correctifs sans temps d'arrêt.

# Mises à jour du moteur de base de données Aurora MySQL du 07/08/2018 (version 1.17.4) (obsolète)

Version : 1.17.4

Aurora MySQL 1.17.4 est en général disponible. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.17.4. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.17.4. Pour utiliser une version antérieure, vous pouvez créer des clusters de bases de données dans Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.3. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17.4 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Améliorations apportées à la réplication :

  - Trafic réseau réduit par la non transmission d'enregistrements de journaux binaires à des réplicas de clusters. Cette améliorations sont activées par défaut.
  - Réduction du trafic réseau en compressant les messages de réplication. Cette amélioration est activée par défaut pour les classes d'instance de 8xlarge et 16xlarge. Ces grandes instances peut gérer un fort volume de trafic en écriture qui entraîne un trafic réseau considérable en termes de messages de réplication.
  - Correctifs du cache de requête de réplica.
- Résolution d'un problème dans lequel `ORDER BY LOWER(col_name)` pouvait produire un ordre incorrect lors de l'utilisation du classement `utf8_bin`.
- Résolution d'un problème dans lequel les instructions DDL (notamment `TRUNCATE TABLE`) pouvaient entraîner des problèmes sur les réplicas Aurora, y compris une instabilité ou des tables manquantes.
- Résolution d'un problème dans lequel les sockets sont laissés dans un état semi-ouvert lorsque les nœuds de stockage sont redémarrés.
- Les nouveaux paramètres de cluster de bases de données suivants sont disponibles :
  - `aurora_enable_zdr` – autorise les connexion ouvertes sur un réplica Aurora à rester actives au redémarrage du réplica.
  - `aurora_enable_replica_log_compression` – autorise la compression des charges utiles de réplication pour améliorer l'utilisation de la bande passante réseau entre le maître et les réplicas Aurora.
  - `aurora_enable_repl_bin_log_filtering` – autorise le filtrage des enregistrements de réplication qui ne peuvent pas être utilisés par les réplicas Aurora sur le maître.

# Mises à jour du moteur de base de données Aurora MySQL du 05/06/2018 (version 1.17.3) (obsolète)

Version : 1.17.3

Aurora MySQL 1.17.3 est en général disponible. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.17.3. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.17.3. Vous pouvez créer de nouveaux clusters de bases de données dans Aurora MySQL 1.14.4, Aurora MySQL 1.15.1 ou Aurora MySQL

1.16. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17.3 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

> **Note**
>
> Cette version n'est actuellement pas disponible dans les régions AWS GovCloud (USA Ouest) [us-gov-west-1] et Chine (Pékin) [cn-north-1]. Sa disponibilité fera l'objet d'une annonce distincte.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

- Correction d'un problème impliquant le redémarrage d'un réplica Aurora lors de l'utilisation des restaurations de curseur optimisé pendant la lecture des enregistrements.
- Correction d'un problème à cause duquel un Aurora Writer redémarrait lorsqu'il essayait de fermer une session MySQL (kill *<session id>* « ») alors que le schéma de performance était activé.
- Correction d'un problème impliquant le redémarrage d'un enregistreur Aurora lors du calcul d'un seuil pour le nettoyage de la mémoire.
- Correction d'un problème impliquant le redémarrage occasionnel d'un enregistreur Aurora lors du suivi de la progression du réplica Aurora dans l'application de journal.
- Correction d'un problème avec le cache de requêtes entraînant des lectures obsolètes lorsque la validation automatique est désactivée.

## Mises à jour du moteur de base de données Aurora MySQL du 27/04/2018 (version 1.17.2) (obsolète)

Version : 1.17.2

Aurora MySQL 1.17.2 est en général disponible. Tous les nouveaux clusters de bases de données Aurora MySQL compatibles avec MySQL 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.17.2. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.17.2. Vous pouvez créer de nouveaux clusters de bases de données dans Aurora MySQL 1.14.4, Aurora MySQL 1.15.1 ou Aurora MySQL 1.16. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17.2 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

- Correction d'un problème entraînant le redémarrage au cours de certaines opérations de partition DDL.
- Correction d'un problème en raison duquel la prise en charge de l'invocation de AWS Lambda fonctions via les fonctions natives d'Aurora MySQL était désactivée.
- Correction d'un problème d'invalidation du cache entraînant le redémarrage sur les réplicas Aurora.
- Correction d'un problème dans le gestionnaire de verrou entraînant des redémarrages.

## Mises à jour du moteur de base de données Aurora MySQL : 23/03/2018 (version 1.17.1) (obsolète)

Version : 1.17.1

Aurora MySQL 1.17.1 est en général disponible. Tous les nouveaux clusters de bases de données, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.17.1. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.17.1. Vous pouvez créer de nouveaux clusters de bases de données dans Aurora MySQL 1.15.1, Aurora MySQL 1.16 ou Aurora MySQL 1.17. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17.1 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Cette version corrige certains problèmes de moteur identifiés, ainsi que certaines régressions.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

> **Note**
>
> Il existe un problème dans la dernière version du moteur Aurora MySQL. Après la mise à niveau vers 1.17.1, la version du moteur a été mentionnée de façon incorrecte comme `1.17`. Si vous avez effectué une mise à niveau vers la version 1.17.1, vous pouvez la confirmer en vérifiant la colonne Maintenance du cluster de bases de données dans AWS Management Console. Si elle affiche `none`, alors le moteur est mis à niveau vers 1.17.1.

### Améliorations

- Correction d'un problème dans la récupération de journal binaire qui se traduisait par des temps de récupération plus longs pour les cas avec des fichiers d'index de journal binaire volumineux à même de se produire si les journaux binaires sont soumis à une rotation très fréquente.
- Correction de problème de l'optimiseur de requête qui générait un plan de requête inefficace pour les tables partitionnées.
- Correction du problème de l'optimiseur de requête où une requête de la plage entraînait un redémarrage du moteur de base de données.

## Mises à jour du moteur de base de données Aurora MySQL du 13/03/2018 (version 1.17) (obsolète)

Version : 1.17

Aurora MySQL 1.17 est en général disponible. Les versions Aurora MySQL 1.x sont uniquement compatibles avec MySQL 5.6 et non avec MySQL 5.7. Tous les nouveaux clusters de bases de données compatibles avec la version 5.6, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora 1.17. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora 1.17. Vous pouvez créer de nouveaux clusters de bases

de données dans Aurora 1.14.1, Aurora 1.15.1 ou Aurora 1.16. Vous pouvez le faire à l'aide de la AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.17 d'Aurora, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Nous prenons en charge l'application de correctifs sans temps d'arrêt, basée sur l'optimisation, afin de conserver les connexions client tout au long du processus de mise à jour corrective. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support.

## Application de correctifs sans temps d'arrêt

La fonction d'application de correctifs sans temps d'arrêt tente, dans un souci d'optimisation, de conserver les connexions client tout au long de la mise à jour corrective du moteur. Pour plus d'informations sur l'application de correctifs sans temps d'arrêt, consultez Utilisation des correctifs sans temps d'arrêt dans le Guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctionnalités

- Aurora MySQL prend désormais en charge la compression de verrou, qui optimise l'utilisation de la mémoire du gestionnaire de verrou. À partir de la version 1.17, vous pouvez utiliser cette fonction sans activer le mode Lab.

## Améliorations

- Correction d'un problème rencontré essentiellement par les instances avec moins de cœurs où un seul cœur peut avoir 100 % d'utilisation d'UC même si la base de données est inactive.
- Amélioration des performances d'extraction des journaux binaires à partir des clusters Aurora.
- Correction d'un problème où les réplicas Aurora tentent d'écrire des statistiques de table sur un stockage permanent, et échouent.
- Correction d'un problème où le cache de requête n'a pas fonctionné comme prévu sur les réplicas Aurora.
- Correction d'une condition de concurrence dans le gestionnaire de verrou qui s'est traduite par un redémarrage du moteur.

- Correction d'un problème où les verrous acceptés par les transactions en lecture seule et à validation automatique se sont traduites par un redémarrage du moteur.
- Correction d'un problème où certaines requêtes n'ont pas été écrites sur les journaux d'audit.
- Correction d'un problème lié à la récupération de certaines opérations de maintenance d'une partition lors d'un basculement.

### Intégration de correctifs de bogues MySQL.

- LAST_INSERT_ID n'est pas répliqué correctement si les filtres de réplication sont utilisés (bogue n ° 69861)
- La requête retourne différents résultats en fonction du paramètre INDEX_MERGE (bogue n ° 16862316)
- Réexécution de la routine stockée du traitement de requête, plan de requête inefficace (bogue n ° 16346367)
- INNODB FTS : assertion dans FTS_CACHE_APPEND_DELETED_DOC_IDS (bogue n° 18079671)
- Assertion RBT_EMPTY(INDEX_CACHE->WORDS) dans ALTER TABLE CHANGE COLUMN (bogue n° 17536995)
- La recherche INNODB en texte intégral ne trouve pas d'enregistrements lorsque des points de sauvegarde sont impliqués (bogue n° 70333, bogue n° 17458835)

## Mises à jour du moteur de base de données Aurora MySQL du 11/12/2017 (version 1.16) (obsolète)

Version : 1.16

Aurora MySQL 1.16 est en disponibilité générale. Tous les nouveaux clusters de bases de données, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora 1.16. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora 1.16. Vous pouvez créer de nouveaux clusters de bases de données dans Aurora 1.14.1 ou Aurora 1.15.1. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.16 d'Aurora, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Nous

activons l'application de correctifs sans temps d'arrêt, basée sur l'optimisation, afin de conserver les connexions client tout au long du processus de mise à jour corrective. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support.

## Application de correctifs sans temps d'arrêt

La fonction d'application de correctifs sans temps d'arrêt tente, dans un souci d'optimisation, de conserver les connexions client tout au long de la mise à jour corrective du moteur. Pour plus d'informations sur l'application de correctifs sans temps d'arrêt, consultez Utilisation des correctifs sans temps d'arrêt dans le Guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctionnalités

- Aurora MySQL prend désormais en charge les AWS Lambda invocations synchrones via la fonction native. `lambda_sync()` La fonction native `lambda_async()` est également disponible et peut être utilisée comme alternative à la procédure stockée existante pour l'appel Lambda asynchrone. Pour plus d'informations, consultez Appel d'une fonction Lambda à partir d'un cluster de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Aurora MySQL prend désormais en charge les jointures par hachage afin d'accélérer les requêtes d'équijointure. L'optimiseur Aurora basé sur les coûts peut décider automatiquement du moment auquel utiliser des jointures par hachage, ou bien vous pouvez imposer leur utilisation dans un plan de requête. Pour plus d'informations, consultez Optimisation des requêtes de jointure MySQL Aurora volumineuses avec des jointures de hachage dans le Guide de l'utilisateur Amazon Aurora.
- Aurora MySQL prend désormais en charge le traitement par lot des analyses afin d'accélérer considérablement les requêtes orientées analyse en mémoire. Cette fonction augmente les performances des analyses complètes de tables, des analyses complètes d'index et des analyses de plages d'index avec un traitement par lot.

## Améliorations

- Correction d'un problème en raison duquel les réplicas en lecture se bloquaient lors de l'exécution de requêtes sur des tables qui venaient d'être supprimées sur l'instance maître.

- Correction d'un problème en raison duquel le redémarrage de l'enregistreur sur un cluster de bases de données avec un très grand nombre d'index FULLTEXT entraîne une récupération plus longue que prévu.
- Correction d'un problème en raison duquel le vidage des journaux binaires entraîne des incidents LOST_EVENTS dans les événements de journaux binaires.
- Correction de problèmes de stabilité avec le planificateur lorsque le schéma de performance est activé.
- Correction d'un problème en raison duquel une sous-requête utilisant des tables temporaires pouvait renvoyer des résultats partiels.

### Intégration de correctifs de bogues MySQL.

Aucun

## Mises à jour du moteur de base de données Aurora MySQL du 20/11/2017 (version 1.15.1) (obsolète)

Version : 1.15.1

Aurora MySQL 1.15.1 est en disponibilité générale. Tous les nouveaux clusters de bases de données, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora 1.15.1. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora 1.15.1. Vous pouvez créer de nouveaux clusters de bases de données dans Aurora 1.14.1. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.15.1 d'Aurora, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Nous activons l'application de correctifs sans temps d'arrêt, basée sur l'optimisation, afin de conserver les connexions client tout au long du processus de mise à jour corrective. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Application de correctifs sans temps d'arrêt

La fonction d'application de correctifs sans temps d'arrêt tente, dans un souci d'optimisation, de conserver les connexions client tout au long de la mise à jour corrective du moteur. Pour plus d'informations sur l'application de correctifs sans temps d'arrêt, consultez Utilisation des correctifs sans temps d'arrêt dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Résolution d'un problème du sélecteur de segment adaptatif pour une requête de lecture qui provoquerait deux fois de suite le choix du même segment, entraînant ainsi un pic de latence de lecture dans certaines conditions.
- Résolution d'un problème issu d'une optimisation dans Aurora MySQL pour le planificateur de thread. Ce problème se manifeste sous la forme d'erreurs intempestives pendant l'écriture dans le journal lent, tandis que les requêtes associées fonctionnent correctement.
- Résolution d'un problème de stabilité lié aux réplicas en lecture sur de grands volumes (> 5 To).
- Résolution d'un problème impliquant l'augmentation continue du nombre de threads de travail en raison d'un faux nombre de connexions restantes.
- Résolution d'un problème de verrous de table qui provoquait de longues attentes de sémaphore pendant les charges de travail d'insert.
- Rétablissement des correctifs de bogues MySQL suivants compris dans Aurora MySQL 1.15 :
  - Ralentissement de l'instance MySQL « doing SYNC index » (bogue n° 73816)
  - Assertion RBT_EMPTY(INDEX_CACHE->WORDS) dans ALTER TABLE CHANGE COLUMN (bogue n° 17536995)
  - La recherche InnoDB Fulltext ne trouve pas d'enregistrements lorsque des points de sauvegarde sont impliqués (bogue n° 70333)

## Intégration de correctifs de bogues MySQL.

Aucun

# Mises à jour du moteur de base de données Aurora MySQL du 24/10/2017 (version 1.15) (obsolète)

Version : 1.15

Aurora MySQL 1.15 est en disponibilité générale. Tous les nouveaux clusters de bases de données, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora 1.15. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora 1.15. Vous pouvez créer de nouveaux clusters de bases de données dans Aurora 1.14.1. Vous pouvez le faire à l'aide de l'API AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur.

Avec la version 1.15 d'Aurora, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Les mises à jour nécessitent un redémarrage de la base de données, si bien que vous rencontrerez 20 à 30 secondes d'indisponibilité, après quoi vous pourrez reprendre l'utilisation de votre ou de vos clusters de bases de données. Si vos clusters de bases de données exécutent actuellement Aurora 1.14 ou Aurora 1.14.1, la fonction d'application de correctifs sans temps d'arrêt d'Aurora MySQL peut permettre aux connexions client à votre instance principale Aurora MySQL de persister tout au long de la mise à niveau, en fonction de votre charge de travail.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Application de correctifs sans temps d'arrêt

La fonction d'application de correctifs sans temps d'arrêt tente, dans un souci d'optimisation, de conserver les connexions client tout au long de la mise à jour corrective du moteur. Pour plus d'informations sur l'application de correctifs sans temps d'arrêt, consultez Utilisation des correctifs sans temps d'arrêt dans le Guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctionnalités

- Lecture anticipée asynchrone des clés – la lecture anticipée asynchrone des clés (AKP) est une fonction qui vise à améliorer les performances des jointures d'index non mises en cache en récupérant au préalable les clés en mémoire avant qu'elles ne soient nécessaires. Le cas d'utilisation principale visé par AKP est une jointure d'index entre une petite table externe et une grande table interne, où l'index de la grande table est très sélectif. Lorsque l'interface Multi-Range Read (MRR) est activée, AKP sera exploité pour une recherche d'index secondaire vers principal. Les instances plus petites avec des contraintes de mémoire peuvent dans certains cas exploiter la fonction AKP selon la cardinalité de clé. Pour plus d'informations, consultez Optimisation des

requêtes de jointure indexées Aurora MySQL avec lecture anticipée asynchrone des clés dans le Guide de l'utilisateur Amazon Aurora.

- Fast DDL – nous avons étendu cette fonction publiée dans Aurora 1.13 aux opérations qui incluent des valeurs par défaut. Avec cette extension, FAST DLL est applicable pour les opérations qui ajoutent une colonne acceptant la valeur null à la fin d'une table, avec ou sans valeurs par défaut. Cette fonction reste sous le mode Lab d'Aurora. Pour plus d'informations, consultez Modification de tables dans Amazon Aurora à l'aide de Fast DDL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'une erreur de calcul lors de l'optimisation des requêtes WITHIN/CONTAINS spatiales qui entraînait auparavant un jeu de résultats vide.
- Résolution du problème associé à la commande `SHOW VARIABLE` afin d'afficher la valeur du paramètre `innodb_buffer_pool_size` mise à jour en cas de modification dans le groupe de paramètres.
- Amélioration de la stabilité de l'instance principale pendant l'insertion en bloc dans une table modifiée avec Fast DDL lorsque l'indexation de hachage est désactivée et lorsque l'enregistrement à insérer est le premier d'une page.
- Amélioration de la stabilité d'Aurora lorsque l'utilisateur tente de définir la valeur du paramètre de cluster de bases de données server_audit_events sur **`default`**.
- Résolution d'un problème où une modification de jeu de caractères de base de données pour une instruction ALTER TABLE exécutée sur l'instance principale d'Aurora n'était pas répliquée sur les réplicas Aurora avant leur redémarrage.
- Amélioration de la stabilité grâce à la correction d'une condition de concurrence sur l'instance principale qui permettait précédemment d'enregistrer un réplica Aurora même si l'instance principale avait fermé son propre volume.
- Amélioration des performances de l'instance principale pendant la création d'index sur une table de grande taille en changeant le protocole de verrouillage pour permettre les instructions DML simultanées pendant la construction d'index.
- Correction du problème d'incohérence des métadonnées InnoDB pendant la requête ALTER TABLE RENAME pour une meilleure stabilité. Exemple : lorsque les colonnes de la table t1(c1, c2) sont renommées cycliquement en t1(c2,c3) dans la même instruction ALTER.
- Amélioration de la stabilité des réplicas Aurora lorsqu'un réplica Aurora n'a pas de charge de travail active et que l'instance principale ne répond pas.

- Amélioration de la disponibilité des réplicas Aurora lorsqu'un réplica Aurora contient un verrou explicite sur une table et empêche le thread de réplication d'appliquer les modifications DDL reçues à partir de l'instance principale.
- Amélioration de la stabilité de l'instance principale lorsqu'une clé étrangère et une colonne sont ajoutées simultanément à une table à partir de deux sessions séparées et que la fonction Fast DDL a été activée.
- Amélioration de la stabilité du thread de purge sur l'instance principale pendant une lourde charge de travail en écriture en bloquant la troncature d'enregistrements d'annulation jusqu'à ce qu'ils soient purgés.
- Amélioration de la stabilité en corrigeant l'ordre de libération du verrou pendant le processus d'engagement des transactions qui ignore des tables.
- Correction d'un défaut lié aux réplicas Aurora dans lequel l'instance de base de données ne pouvait par terminer le démarrage et indiquait que le port 3306 était déjà utilisé.
- Correction d'une condition de concurrence dans laquelle l'exécution d'une requête SELECT sur certaines tables information_schema (innodb_trx, innodb_lock, innodb_lock_waits) augmentait l'instabilité du cluster.

## Intégration de correctifs de bogues MySQL.

- CREATE USER accepte le plugin et le hachage du mot de passe, mais ignore le hachage du mot de passe (bogue n° 78033)
- Le moteur de partitionnement ajoute des champs à l'ensemble de bits de lecture pour pouvoir retourner des entrées triées à partir d'un index partitionné. Par conséquent, le tampon de jointure essaie de lire des champs non nécessaires. Corrigé en n'ajoutant pas tous les champs de partitionnement à read_set, mais en triant uniquement les champs à préfixe déjà spécifié de read_set. Ajout d'un DBUG_ASSERT qui implique la lecture du premier champ en cas de key_cmp (bogue n° 16367691).
- Ralentissement de l'instance MySQL « doing SYNC index » (bogue n° 73816)
- Assertion RBT_EMPTY(INDEX_CACHE->WORDS) dans ALTER TABLE CHANGE COLUMN (bogue n° 17536995)
- La recherche InnoDB Fulltext ne trouve pas d'enregistrements lorsque des points de sauvegarde sont impliqués (bogue n° 70333)

# Mises à jour du moteur de base de données Aurora MySQL : 13/03/2018 (version 1.14.4) (obsolète)

Version : 1.14.4

Aurora MySQL 1.14.4 est en disponibilité générale. Vous pouvez créer de nouveaux clusters de bases de données dans Aurora 1.14.4, à l'aide de la AWS CLI ou de l'API Amazon RDS et en spécifiant la version du moteur. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données 1.14.x vers Aurora 1.14.4.

Avec la version 1.14.4 d'Aurora, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Nous prenons en charge l'application de correctifs sans temps d'arrêt, basée sur l'optimisation, afin de conserver les connexions client tout au long du processus de mise à jour corrective. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Application de correctifs sans temps d'arrêt

La fonction d'application de correctifs sans temps d'arrêt tente, dans un souci d'optimisation, de conserver les connexions client tout au long de la mise à jour corrective du moteur. Pour plus d'informations sur l'application de correctifs sans temps d'arrêt, consultez Utilisation des correctifs sans temps d'arrêt dans le Guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctionnalités

- Aurora MySQL prend désormais en charge les classes d'instance db.r4.

## Améliorations

- Correction d'un problème où `LOST_EVENTS` étaient générés lors de l'écriture d'événements de journaux binaires volumineux.

## Intégration de correctifs de bogues MySQL.

- Les événements pouvant être ignorés ne fonctionnent pas et ne sont pas testés (bogue n° 74683)
- NEW->OLD ASSERT FAILURE 'GTID_MODE > 0' (Bug #20436436)

# Mises à jour du moteur de base de données Aurora MySQL : 22/09/2017 (version 1.14.1) (obsolète)

Version : 1.14.1

Aurora MySQL 1.14.1 est en disponibilité générale. Tous les nouveaux clusters de bases de données, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.14.1. Aurora MySQL 1.14.1 est également une mise à niveau obligatoire pour les clusters de bases de données Aurora MySQL existants. Pour plus d'informations, consultez Annonce : extension du calendrier de mise à niveau obligatoire pour Amazon Aurora sur le site Web des forums pour AWS développeurs.

Avec la version 1.14.1 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora MySQL sont corrigés en même temps. Les mises à jour nécessitent un redémarrage de la base de données, si bien que vous rencontrerez 20 à 30 secondes d'indisponibilité, après quoi vous pourrez reprendre l'utilisation de votre ou de vos clusters de bases de données. Si vos clusters de bases de données exécutent actuellement la version 1.13 ou ultérieure, la fonction d'application de correctifs sans temps d'arrêt d'Aurora MySQL peut permettre aux connexions client à votre instance principale Aurora MySQL de persister tout au long de la mise à niveau, en fonction de votre charge de travail.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support.

## Améliorations

- Correction de conditions de concurrence associées aux insertions et purge pour améliorer la stabilité de la fonction Fast DDL qui reste en mode Lab Aurora MySQL.

# Mises à jour du moteur de base de données Aurora MySQL : 07/08/2017 (version 1.14) (obsolète)

Version : 1.14

Aurora MySQL 1.14 est en général disponible. Tous les nouveaux clusters de bases de données, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.14. Aurora MySQL 1.14 est également une mise à niveau obligatoire pour les clusters de bases de données Aurora MySQL existants. Nous enverrons une annonce séparée avec le calendrier de prise en charge des versions précédentes d'Aurora MySQL.

Avec la version 1.14 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Les mises à jour nécessitent un redémarrage de la base de données, si bien que vous rencontrerez 20 à 30 secondes d'indisponibilité, après quoi vous pourrez reprendre l'utilisation de votre ou de vos clusters de bases de données. Si vos clusters de bases de données exécutent actuellement la version 1.13, la fonction d'application de correctifs sans temps d'arrêt d'Aurora peut permettre aux connexions client à votre instance principale Aurora de persister tout au long de la mise à niveau, en fonction de votre charge de travail.

Si vous avez des questions ou des préoccupations, le AWS support est disponible sur les forums communautaires et via le AWS support.

## Application de correctifs sans temps d'arrêt

La fonction d'application de correctifs sans temps d'arrêt tente, dans un souci d'optimisation, de conserver les connexions client tout au long de la mise à jour corrective du moteur. Pour plus d'informations sur l'application de correctifs sans temps d'arrêt, consultez Utilisation des correctifs sans temps d'arrêt dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'une erreur incorrecte « Enregistrement introuvable » lorsqu'un enregistrement est trouvé dans l'index secondaire, mais pas dans l'index principal.
- Correction d'un problème de stabilité susceptible de se produire suite à une assertion de défense (ajoutée dans la version 1.12) qui était trop forte dans le cas où une écriture individuelle s'étend sur 32 pages. Une telle situation peut se produire, par exemple, avec des valeurs BLOB élevées.

- Correction d'un problème de stabilité à cause des incohérences entre le cache de l'espace disque logique et le cache de dictionnaire.
- Correction d'un problème au cours duquel le réplica Aurora cesse de réagir après avoir dépassé le nombre maximal de tentatives de connexion à l'instance principale. Un réplica Aurora redémarre maintenant si la période d'inactivité est supérieure à la période de pulsation utilisée pour la vérification de l'état par l'instance principale.
- Correction d'un blocage opérationnel susceptible de se produire à la suite d'actions simultanées lorsqu'une connexion tente d'acquérir un blocage de métadonnées exclusif au moment de l'émission d'une commande, telle que `ALTER TABLE`.
- Correction d'un problème de stabilité dans une réplique d'Aurora Read en présence de logical/parallel Read Ahead.
- Amélioration de `LOAD FROM S3` de deux façons :
  1. Meilleure gestion des erreurs de délai d'attente Amazon S3 à l'aide de la nouvelle tentative du kit SDK en plus de la tentative existante.
  2. Optimisation des performances lors de la charge de très gros fichiers ou de grands nombres de fichiers par la mise en cache et la réutilisation de l'état du client.
- Correction des problèmes de stabilité suivants avec Fast DDL pour les opérations `ALTER TABLE` :
  1. Lorsque l'instruction `ALTER TABLE` comporte plusieurs commandes `ADD COLUMN` et que les noms de colonne ne sont pas dans l'ordre croissant.
  2. Lorsque la chaîne de nom de la colonne à mettre à jour et sa chaîne de nom correspondante, extraites à partir de la table système interne, varient en fonction d'un caractère de fin null (/0).
  3. Sous certaines opérations de fractionnement de l'arbre B (B-tree).
  4. Lorsque la table comporte une clé primaire de longueur variable.
- Correction d'un problème de stabilité avec les réplicas Aurora lorsqu'il faut trop de temps pour rendre le cache d'index de recherche en texte intégral cohérent avec celui de l'instance principale. Cela peut se produire si une grande partie des entrées d'index de recherche en texte intégral récemment créées sur l'instance principale n'a pas encore été vidée sur le disque.
- Correction d'un problème de stabilité susceptible de se produire pendant la création de l'index.
- Nouvelle infrastructure qui suit la consommation de mémoire par connexion et télémétrie associée qui sera utilisée pour élaborer des stratégies d'évitement Out-Of-Memory (OOM).
- Correction d'un problème où `ANALYZE TABLE` était incorrectement autorisé sur les réplicas Aurora. Ceci est maintenant bloqué.

- Correction d'un problème de stabilité causé par un blocage rare suite à une condition rare entre la lecture anticipée logique et la purge.

### Intégration de correctifs de bogues MySQL.

- Une recherche en texte intégral associée aux tables dérivées (sous-requêtes de la clause `FROM`) entraînait un arrêt du serveur. Maintenant, si une opération de texte intégral dépend d'une table dérivée, le serveur produit une erreur indiquant qu'une recherche en texte intégral ne peut pas être effectuée sur une table matérialisée. (Bogues n° 68751 et 16539903)

## Mises à jour du moteur de base de données Aurora MySQL : 15/05/2017 (version 1.13) (obsolète)

Version : 1.13

> **Note**
>
> Nous avons activé une nouvelle fonction - SELECT INTO OUTFILE S3 - dans la version 1.13 d'Aurora MySQL après la version initiale et mis à jour les notes de mise à jour de manière à refléter cette modification.

Aurora MySQL 1.13 est en général disponible. Tous les nouveaux clusters de bases de données, y compris ceux restaurés à partir d'instantanés, seront créés dans Aurora MySQL 1.13. Vous avez la possibilité, sans y être obligé, de mettre à niveau les clusters de bases de données existants vers Aurora MySQL 1.13. Avec la version 1.13 d'Aurora, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Nous activons l'application de correctifs sans temps d'arrêt, basée sur l'optimisation, afin de conserver les connexions client tout au long du processus de mise à jour corrective. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

### Application de correctifs sans temps d'arrêt

La fonction d'application de correctifs sans temps d'arrêt tente, dans un souci d'optimisation, de conserver les connexions client tout au long de la mise à jour corrective du moteur. Pour plus

d'informations sur l'application de correctifs sans temps d'arrêt, consultez Utilisation des correctifs sans temps d'arrêt dans le Guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctions :

- SELECT INTO OUTFILE S3 – Aurora MySQL vous permet maintenant de charger les résultats d'une requête dans un ou plusieurs fichiers d'un compartiment Amazon S3. Pour plus d'informations, consultez Enregistrement de données d'un cluster de bases de données Amazon Aurora MySQL dans des fichiers texte stockés dans un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations :

- La troncation des fichiers journaux au format CSV au démarrage du moteur a été mise en œuvre pour éviter une longue durée de récupération. Désormais, les tables `general_log_backup`, `general_log`, `slow_log_backup` et `slow_log` ne survivent pas à un redémarrage de la base de données.
- Correction d'un problème d'échec de la migration d'une base de données nommée **`test`**.
- La stabilité au niveau du récupérateur de mémoire du gestionnaire de verrous a été améliorée en réutilisant les segments de verrouillage corrects.
- La stabilité du gestionnaire de verrous a été améliorée par la suppression des assertions non valides au cours de l'exécution de l'algorithme de détection de blocage.
- La réplication asynchrone a été réactivée et un problème associé a été corrigé qui entraînait un retard du réplica incorrect à signaler sous une charge de travail nulle ou en lecture seule. Les améliorations du pipeline de réplication ont été introduites dans la version 1.10. Ces améliorations ont été introduites afin d'appliquer les mises à jour des flux de journaux au cache des tampons d'un réplica Aurora, ce qui permet d'améliorer les performances de lecture et la stabilité sur les réplicas Aurora.
- Un problème a été corrigé où autocommit=OFF conduisait au blocage d'événements planifiés et au maintien de longues transactions ouvertes jusqu'au redémarrage du serveur.
- Un problème a été corrigé où les journaux des requêtes générales, d'audit et lentes ne pouvaient pas consigner les requêtes traitées par la validation asynchrone.
- Les performances de la fonction de lecture anticipée logique ont été améliorées jusqu'à 2,5 fois. Cela s'est fait en autorisant la poursuite des récupérations au préalable sur les pages intermédiaires dans une arborescence B.

- La validation des paramètres a été ajoutée pour que les variables d'audit suppriment les espaces inutiles.
- Une régression a été corrigée, qui avait été introduite dans Aurora MySQL version 1.11, en raison de laquelle les requêtes pouvaient retourner des résultats incorrects lors de l'utilisation de l'option SQL_CALC_FOUND_ROWS et de l'appel de la fonction FOUND_ROWS().
- Un problème de stabilité a été corrigé qui correspondait à l'établissement incorrect de la liste des verrous de métadonnées.
- La stabilité est améliorée lorsque sql_mode a pour valeur PAD_CHAR_TO_FULL_LENGTH et que la commande `SHOW FUNCTION STATUS WHERE Db='string'` est exécutée.
- Un cas rare a été corrigé où les instances n'étaient pas démarrées après la mise à niveau de la version d'Aurora en raison d'un contrôle de cohérence de volume erroné.
- Correction du problème de performances, introduit dans Aurora MySQL version 1.12, où les performances de l'enregistreur Aurora étaient réduites lorsque les utilisateurs possédaient un grand nombre de tables.
- Amélioration du problème de stabilité lorsque l'enregistreur Aurora est configuré en tant que travailleur des journaux binaires et que le nombre de connexions approche les 16 000.
- Correction d'un problème rare où un réplica Aurora pouvait redémarrer lorsqu'une connexion était bloquée en attente d'un verrou de métadonnées lors de l'exécution d'une DDL sur le maître Aurora.

## Intégration de correctifs de bogues MySQL.

- Avec une table InnoDB vide, il est impossible de diminuer la valeur auto_increment à l'aide d'une instruction ALTER TABLE, même lorsque la table est vide. (Bogue n° 69882)
- Les requêtes MATCH()... AGAINST utilisant une longue chaîne comme argument pour AGAINST() pouvaient entraîner une erreur lorsqu'elles étaient exécutées sur une table InnoDB avec un index de recherche en texte intégral. (Bogue n° 17640261)
- Le traitement de SQL_CALC_FOUND_ROWS en association avec ORDER BY et LIMIT pouvait mener à des résultats incorrects pour FOUND_ROWS(). (Bogues n° 68458 et 16383173)
- ALTER TABLE n'autorise pas la modification de la possibilité de valeur NULL de la colonne si une clé étrangère existe. (Bogue n° 77591)

# Mises à jour du moteur de base de données Aurora MySQL : 05/04/2017 (version 1.12) (obsolète)

Version : 1.12

Aurora MySQL 1.12 constitue désormais la version conseillée pour la création de nouveaux clusters de bases de données, y compris les restaurations à partir d'instantanés.

Il n'est pas obligatoire d'effectuer la mise à niveau pour les clusters existants. Vous pourrez mettre à niveau les clusters existants vers la version 1.12 une fois que nous aurons terminé d'appliquer le correctif à l'échelle du parc à la version 1.11 (consultez les notes de mise à jour d'Aurora 1.11 et l'annonce du forum correspondante). Avec la version 1.12 d'Aurora, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Nouvelles fonctionnalités

- Fast DDL – Aurora MySQL vous permet désormais d'exécuter une opération ALTER TABLE tbl_name ADD COLUMN col_name column_definition presque instantanément. L'opération s'effectue sans nécessiter la copie de la table et sans impact matériel sur les autres instructions DML. Puisque l'opération ne consomme pas de stockage temporaire pour une copie de table, les instructions DDL sont pratiques même pour des tables volumineuses sur des classes d'instance Small. FAST DLL prend actuellement uniquement en charge l'ajout d'une colonne acceptant la valeur null, sans valeur par défaut, à la fin d'une table. Cette fonction n'est actuellement disponible qu'au mode Lab d'Aurora. Pour plus d'informations, consultez Modification de tables dans Amazon Aurora à l'aide de Fast DDL dans le Guide de l'utilisateur Amazon Aurora.
- Show volume status – Nous avons ajouté une nouvelle commande de surveillance, SHOW VOLUME STATUS, qui permet d'afficher le nombre de nœuds et de disques d'un volume. Pour plus d'informations, consultez Affichage du statut du volume pour un cluster de base de données Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Modifications implémentées pour verrouiller la compression en vue de réduire davantage la mémoire allouée par objet verrouillé. Cette amélioration est disponible au mode Lab.

- Un problème de diminution rapide de la métrique `trx_active_transactions`, même quand la base de données est inactive, a été résolu.
- Un problème de message d'erreur non valide concernant la syntaxe de la requête d'injection d'erreurs en cas d'échec de la simulation dans les disques et les nœuds a été résolu.
- Plusieurs problèmes de conditions de concurrence et de verrous morts dans le gestionnaire de verrous ont été résolus.
- Un problème de dépassement de mémoire tampon dans l'optimiseur de requête a été résolu.
- Un problème de stabilité dans les réplicas en lecture de l'Aurora lorsque les nœuds de stockage sous-jacents ont une mémoire disponible faible a été résolu.
- Un problème de persistance de connexions inactives au-delà de la définition du paramètre `wait_timeout` a été résolu.
- Un problème de renvoi par `query_cache_size` d'une valeur inattendue après le redémarrage d'une instance a été résolu.
- Un problème de performance résultant d'un fil de diagnostic détectant le réseau trop souvent dans le cas où les écritures ne progressent pas vers le stockage a été résolu.

## Intégration de correctifs de bogues MySQL.

- Le rechargement d'une table qui a été expulsée quand elle était vide a provoqué la réinitialisation d'une valeur AUTO_INCREMENT. (Bogues n° 21454472 et 77743)
- Un enregistrement d'index a été introuvable à la restauration en raison d'incohérences dans la structure purge_node_t. Celles-ci ont entraîné des messages d'avertissement et d'erreur tels que « erreur dans la mise à jour de l'entrée d'index secondaire », « impossible de purger un enregistrement » et « tentative de purge de l'entrée d'index secondaire non marquée pour la suppression ». (Bogues n° 19138298, 70214, 21126772 et 21065746)
- Le calcul incorrect de la taille de la pile de l'opération qsort entraîne un dépassement de la pile. (Bogue n° 73979)
- Enregistrement introuvable dans un index à la restauration. (Bogues n° 70214 et 72419)
- ALTER TABLE ajoute la colonne TIMESTAMP sur la mise à jour CURRENT_TIMESTAMP insère des données ZERO (Bogue n° 17392)

# Mises à jour du moteur de base de données Aurora MySQL : 23/02/2017 (version 1.11) (obsolète)

Version : 1.11

Nous corrigerons tous les clusters de bases de données Aurora MySQL avec la dernière version sur une courte période suivant la sortie. Les clusters de bases de données sont corrigés grâce à la procédure existante avec un temps d'arrêt d'environ 5 à 30 secondes.

La correction se déroule pendant la fenêtre de maintenance du système que vous avez spécifiée pour chacune de vos instances de base de données. Vous pouvez consulter ou modifier cette fenêtre grâce à l AWS Management Console. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

Vous pouvez également appliquer le correctif immédiatement dans le en AWS Management Console choisissant un cluster de base de données, en choisissant Cluster Actions, puis en choisissant Upgrade Now.

Avec la version 1.11 d'Aurora MySQL, nous utilisons un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster de bases de données Aurora sont corrigés en même temps.

## Nouvelles fonctions

- Option MANIFEST de la commande LOAD DATA FROM S3 – La commande LOAD DATA FROM S3 a été introduite dans la version 1.8. Les options de cette commande ont été développées et vous pouvez désormais spécifier une liste de fichiers à charger dans un cluster de bases de données Aurora à partir d'Amazon S3 à l'aide d'un fichier manifeste. Cela facilite le chargement des données à partir de fichiers spécifiques dans un ou plusieurs emplacements, comparé au chargement des données à partir d'un seul fichier grâce à l'option FILE ou au chargement des données à partir de plusieurs fichiers ayant le même emplacement et préfixe grâce à l'option PREFIX. Le format du fichier manifest est identique à celui utilisé par Amazon Redshift. Pour plus d'informations sur l'utilisation de LOAD DATA FROM S3 avec l'option MANIFEST, consultez Utilisation d'un manifeste pour spécifier les fichiers de données à charger dans le Guide de l'utilisateur Amazon Aurora.
- Indexation spatiale activée par défaut – Cette fonction est sortie en mode Lab dans la version 1.10 et est désormais activée par défaut. L'indexation spatiale améliore les performances des requêtes sur des jeux de données volumineux pour les requêtes qui utilisent des données spatiales. Pour

plus d'informations sur l'utilisation de l'indexation spatiale, consultez Amazon Aurora MySQL et données spatiales dans le Guide de l'utilisateur Amazon Aurora.

- Modification de la durée de l'audit avancé – Cette fonction est sortie dans la version 1.10.1 pour fournir une installation hautement performante afin de réaliser l'audit de l'activité de la base de données. Dans cette version, la précision des horodatages de journaux d'audit a été modifiée d'une seconde à une microseconde. Les horodatages plus précis vous permettent de mieux comprendre quand un événement d'audit se produit. Pour plus d'informations sur l'audit, consultez Utilisation de l'Audit avancé avec un cluster de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Modification du paramètre `thread_handling` pour vous empêcher de lui affecter une autre valeur que **`multiple-connections-per-thread`**, qui est le seul modèle pris en charge avec le groupe de threads d'Aurora.
- Résolution d'un problème causé lorsque vous configurez le paramètre `buffer_pool_size` ou `query_cache_size` pour qu'il dépasse la mémoire totale du cluster de bases de données. Dans ces cas-là, Aurora configure le paramètre modifié sur la valeur par défaut, afin que le cluster de bases de données puisse démarrer sans tomber en panne.
- Résolution d'un problème dans le cache de requête où une transaction obtient des résultats de lecture obsolètes si la table est invalidée dans une autre transaction.
- Résolution d'un problème où des fichiers de journaux binaires à supprimer sont effacés après un court délai au lieu d'être supprimés immédiatement.
- Résolution d'un problème où une base de données créée avec le nom **`tmp`** est traitée comme une base de données système stockée sur un stockage éphémère au lieu d'être conservée dans un stockage distribué Aurora.
- Modification du comportement de SHOW TABLES pour exclure certaines tables système internes. Cette modification permet d'éviter un basculement inutile causé par le verrouillage par mysqldump de tous les fichiers affichés dans SHOW TABLES, qui à son tour empêche les écritures sur la table système interne causées par le basculement.
- Résolution d'un problème où un réplica Aurora redémarre de manière incorrecte lors de la création d'une table temporaire à partir d'une requête qui invoque une fonction dont l'argument est une colonne d'une table InnoDB.

- Résolution d'un problème lié à un conflit de verrouillage de métadonnées dans un nœud de réplica Aurora qui fait prendre du retard au réplica Aurora par rapport au cluster de bases de données principal et entraîne finalement son redémarrage.
- Réparation d'un verrou mort dans le pipeline de réplication des nœuds du lecteur qui fait prendre du retard au réplica Aurora pour finalement être redémarré.
- Résolution d'un problème où un réplica Aurora se décale trop avec les volumes chiffrés supérieurs à 1 téraoctet (To).
- Amélioration de la détection des verrous morts du réplica Aurora grâce à un meilleur moyen de lire l'horloge système.
- Résolution d'un problème où un réplica Aurora peut redémarrer deux fois au lieu d'une suite à l'annulation de l'enregistrement par l'enregistreur.
- Résolution d'un problème de performances de requête lentes sur des réplicas Aurora qui se produisent lorsque les statistiques transitoires entraînent des écarts de statistiques sur les colonnes d'index non uniques.
- Résolution d'un problème où un réplica Aurora peut tomber en panne lorsqu'une instruction DDL est répliquée sur le réplica Aurora alors que ce réplica Aurora traite une requête connexe.
- Modification des améliorations du pipeline de réplication présentées dans la version 1.10 et activées par défaut qui sont désormais désactivées par défaut. Ces améliorations ont été introduites afin d'appliquer des mises à jour de flux de journaux au cache des tampons d'un réplica Aurora, et bien que cette fonction permette d'améliorer les performances de lecture et la stabilité des réplicas Aurora, elle augmente le retard du réplica dans certaines charges de travail.
- Résolution d'un problème où l'occurrence simultanée d'une instruction DDL continue et de la lecture anticipée parallèle en attente sur une même table cause un échec d'assertion lors de la phase de validation de la transaction DDL.
- Amélioration du journal général et du journal des requêtes lentes pour survivre au redémarrage du cluster de bases de données.
- Correction d'un out-of-memory problème lié à certaines requêtes de longue durée en réduisant la consommation de mémoire dans le module ACL.
- Résolution d'un problème de redémarrage qui se produit lorsqu'une table possède des index non spatiaux, lorsque des prédicats spatiaux se trouvent dans la requête, lorsque le planificateur choisit d'utiliser un index non spatial et lorsque le planificateur transmet de manière incorrecte la condition spatiale à l'index.

- Correction d'un problème à cause duquel le cluster de base de données redémarre en cas de suppression, de mise à jour ou de purge d'objets géospatiaux de très grande taille stockés en externe (par exemple LOBs).
- Correction d'un problème de simulation de panne avec ALTER SYSTEM SIMULATE ... FOR INTERVAL ne fonctionne pas correctement.
- Résolution d'un problème de stabilité causé par une assertion non valide sur un invariant incorrect dans le gestionnaire de verrous.
- Désactivation des deux améliorations suivantes dans InnoDB Full-Text Search présentées dans la version 1.10 car elles causaient des problèmes de stabilité pour les charges de travail exigeantes :
  - Mise à jour du cache uniquement après une demande de lecture à un réplica Aurora afin d'améliorer la vitesse de réplication de cache d'index de recherche en texte intégral.
  - Déchargement de la tâche de synchronisation du cache sur un thread séparé dès que la taille du cache dépasse 10 % de la taille totale, afin d'éviter que les requêtes MySQL ne calent trop longtemps au cours de la synchronisation du cache FTS sur le disque. (Bogues n° 22516559, n ° 73816).

### Intégration de correctifs de bogues MySQL.

- L'exécution d'une clé étrangère DROP de table ALTER simultanément avec une autre opération DROP fait disparaître la table. (Bogue n° 16095573)
- Certaines requêtes INFORMATION_SCHEMA qui utilisaient ORDER BY n'utilisaient pas d'optimisation filesort comme elles le faisaient auparavant. (Bogue n° 16423536)
- FOUND_ROWS () renvoie le mauvais nombre de lignes sur une table. (Bogue n° 68458)
- Le serveur échoue au lieu d'indiquer une erreur lorsque trop de tables temporaires sont ouvertes. (Bogue n° 18948649)

## Mises à jour du moteur de base de données Aurora MySQL : 12/01/2017 (version 1.10.1) (obsolète)

Version : 1.10.1

La version 1.10.1 d'Aurora MySQL est une version d'acceptation qui n'est pas utilisée pour corriger vos instances de bases de données. Elle est disponible pour créer des instances Aurora et mettre à niveau les instances existantes. Vous pouvez appliquer le correctif en choisissant un cluster dans

la console Amazon RDS, puis Actions de clusters et Mettre à niveau maintenant. Pour appliquer un correctif, vous devez redémarrer la base de données en respectant un temps d'arrêt de 5 à 30 secondes en général, après quoi vous pouvez de nouveau utiliser vos clusters de bases de données. Ce correctif utilise un modèle d'application de correctifs de cluster dans lequel tous les nœuds d'un cluster Aurora sont corrigés en même temps.

### Nouvelles fonctions

- Audit avancé – Aurora MySQL propose une fonction d'audit avancé hautes performances que vous pouvez utiliser pour effectuer un audit de l'activité de la base de données. Pour plus d'informations sur l'activation et l'utilisation de l'audit avancé, consultez la section Utilisation de l'Audit avancé avec un cluster de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

### Améliorations

- Résolution d'un problème avec l'indexation spatiale lorsque vous créez une colonne et que vous y ajoutez un index dans la même instruction.
- Résolution d'un problème dans lequel les statistiques spatiales ne sont pas conservées après le redémarrage du cluster de bases de données.

## Mises à jour du moteur de base de données Aurora MySQL : 14/12/2016 (version 1.10) (obsolète)

Version : 1.10

### Nouvelles fonctions

- Application de correctifs sans temps d'arrêt – Cette fonction permet d'appliquer des correctifs à une instance de base de données sans aucun temps d'arrêt. Autrement dit, les mises à niveau de la base de données sont effectuées sans déconnecter les applications clientes ni redémarrer la base de données. Cette approche augmente la disponibilité de vos clusters de bases de données Aurora pendant la fenêtre de maintenance. Notez que les données temporaires comme celles dans le schéma de performances sont réinitialisées au cours du processus de mise à niveau. Cette fonction s'applique aux correctifs fournis par le service au cours d'une fenêtre de maintenance, ainsi qu'aux correctifs initiés par l'utilisateur.

Lorsqu'un correctif est démarré, le service s'assure qu'il n'existe aucun verrou ouvert, aucune transaction ni aucune table temporaire, puis attend une fenêtre propice à la correction et au redémarrage de la base de données. Les sessions d'application sont conservées, même en cas de baisse de débit pendant l'application des correctifs (pendant environ 5 secondes). Si aucune fenêtre adaptée n'est disponible, l'application des correctifs correspond par défaut au comportement standard d'application des correctifs.

L'application des correctifs sans temps d'arrêt se déroule autant que faire se peut, sous certaines restrictions, comme décrit ci-dessous :

- Cette fonction est actuellement en vigueur pour l'application des correctifs aux clusters de bases de données à un seul nœud ou aux instances de l'enregistreur dans des clusters de bases de données à plusieurs nœuds.
- Les connexions SSL ne sont pas prises en charge conjointement à cette fonction. S'il existe des connexions SSL actives, Amazon Aurora MySQL n'appliquera pas de correctif sans temps d'arrêt et, à la place, réessaiera régulièrement de voir si les connexions SSL ont pris fin. Si tel est le cas, l'application de correctifs sans temps d'arrêt a lieu. Si les connexions SSL persistent plus de quelques secondes, l'application de correctifs standard avec temps d'arrêt a lieu.
- La fonction est disponible dans Aurora 1.10 et versions ultérieures. Nous allons désormais identifier toutes les versions ou correctifs qui ne peuvent pas être appliqués via l'application de correctifs sans temps d'arrêt.
- Cette fonction n'est pas applicable si la réplication basée sur la journalisation binaire est active.

- Indexation spatiale – L'indexation spatiale améliore les performances des requêtes sur les jeux de données volumineux pour les requêtes qui utilisent des données spatiales. Pour plus d'informations sur l'utilisation de l'indexation spatiale, consultez Amazon Aurora MySQL et données spatiales dans le Guide de l'utilisateur Amazon Aurora.

  Cette fonction est désactivée par défaut et peut être activée en activant le mode Lab d'Aurora. Pour plus d'informations, consultez Mode Lab Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

- Améliorations du pipeline de réplication – Aurora MySQL utilise désormais un mécanisme amélioré pour appliquer les mises à jour des flux de journaux au cache des tampons d'un réplica Aurora. Cette fonction améliore les performances en lecture et la stabilité sur les réplicas Aurora en cas de lourde charge en écriture sur le maître et d'une charge en lecture importante sur le réplica. Cette caractéristique est activée par défaut.

- Amélioration du débit pour les charges de travail avec des lectures mises en cache – Aurora MySQL utilise désormais un algorithme simultané sans verrou pour implémenter les vues en lecture, ce qui permet un meilleur débit pour les requêtes de lecture traitées par le cache de tampons. Grâce à ces améliorations et à d'autres, Amazon Aurora MySQL peut atteindre un débit de 625 000 lectures par seconde, contre 164 000 lectures par seconde pour MySQL 5.7 pour une charge de travail basée uniquement sur la sélection. SysBench
- Amélioration du débit pour les charges de travail avec des conflits entre les lignes critiques – Aurora MySQL utilise un nouvel algorithme de publication de verrou qui améliore les performances, en particulier en cas de conflit de page critique (c'est-à-dire lorsque de nombreuses transactions sont en conflit pour les lignes de la même page). Dans les tests de référence TPC-C, cela peut entraîner une amélioration du débit permettant de multiplier par 16 le nombre de transactions par minute par rapport à MySQL 5.7. Cette fonction est désactivée par défaut et peut être activée en activant le mode Lab d'Aurora. Pour plus d'informations, consultez Mode Lab Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- La vitesse de réplication du cache d'index de recherche en texte intégral a été augmentée en mettant à jour le cache uniquement après une demande de lecture adressée à un réplica Aurora. Cette approche évite les lectures du disque par le thread de réplication.
- Correction d'un problème où l'invalidation du cache de dictionnaire ne fonctionne pas sur un réplica Aurora pour les tables qui ont un caractère spécial dans le nom de base de données ou le nom de table.
- Correction d'un problème `STUCK IO` lors de la migration des données pour les nœuds de stockage distribués lorsque la gestion de la chaleur de stockage est activée.
- Correction d'un problème dans le gestionnaire de verrous, alors qu'une vérification d'assertion échoue pour le thread d'attente de verrouillage de transaction lors de la préparation de la restauration ou de la validation d'une transaction.
- Correction d'un problème lors de l'ouverture d'une table de dictionnaire endommagée en mettant à jour correctement le nombre de référence dans les entrées de la table de dictionnaire.
- Correction d'un bogue selon lequel le point de lecture minimal du cluster de bases de données peut être détenu par des réplicas Aurora lents.
- Correction d'une fuite de mémoire potentielle dans le cache de requête.
- Correction d'un bogue selon lequel un réplica Aurora place un verrou de niveau de ligne sur une table lorsqu'une requête est utilisée dans une instruction `IF` d'une procédure stockée.

### Intégration de correctifs de bogues MySQL.

- L'UNION de tables dérivées retourne des résultats erronés avec des clauses '1=0/false'. (Bogue n ° 69471)
- Le serveur se bloque dans ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS à la 2e exécution de la procédure stockée. (Bogue n° 20755389)
- Empêchez les requêtes MySQL de caler trop longtemps au cours de la synchronisation du cache FTS sur le disque en déchargeant la tâche de synchronisation du cache sur un thread séparé, dès que la taille du cache dépasse 10 % de la taille totale. (Bogue n° 22516559, n° 73816)

## Mises à jour du moteur de base de données Aurora MySQL : 10/11/2016 (versions 1.9.0, 1.9.1) (obsolète)

Version : 1.9.0, 1.9.1

### Nouvelles fonctions

- Construction d'index améliorée – La création d'index secondaires fonctionne désormais suivant un mode de construction de bas en haut, qui élimine les fractionnements de pages inutiles. Cela peut réduire le temps nécessaire à la création d'un index ou la reconstruction d'une table de 75 % (sur la base d'une instance de base de données `db.r3.8xlarge`). Cette fonction était en mode Lab dans Aurora MySQL version 1.7 et est activée par défaut dans Aurora 1.9 et versions ultérieures. Pour plus d'informations, consultez Mode Lab Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Compression de verrou (mode Lab) – Cette implémentation réduit de manière significative la quantité de mémoire consommée par le gestionnaire de verrous, jusqu'à 66 %. Le gestionnaire de verrous peut acquérir davantage de verrous de ligne sans rencontrer d' out-of-memoryexception. Cette fonction est désactivée par défaut et peut être activée en activant le mode Lab d'Aurora. Pour plus d'informations, consultez Mode Lab Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Schéma de performances – Aurora MySQL prend désormais en charge le schéma de performances avec un impact minimal sur les performances. Lors de nos tests d'utilisation SysBench, l'activation du schéma de performance pourrait dégrader les performances de MySQL jusqu'à 60 %.

SysBench les tests d'un cluster de base de données Aurora ont montré un impact sur les performances 4 fois inférieur à celui de MySQL. L'exécution de la classe d'`db.r3.8xlarge`instance de base de données a généré 100 000 requêtes SQL writes/sec et plus de 550 000 lectures SQL par seconde, même avec le schéma de performance activé.

- Amélioration des conflits entre les lignes critiques – Cette fonction réduit l'utilisation de l'UC et augmente le débit lorsqu'un grand nombre de connexions accèdent à un petit nombre de lignes critiques. Cette fonction élimine également l' `error 188` en cas de conflits entre les lignes critiques.
- out-of-memoryGestion améliorée : lorsque des instructions SQL verrouillables non essentielles sont exécutées et que le pool de mémoire réservé est violé, Aurora force l'annulation de ces instructions SQL. Cette fonctionnalité permet de libérer de la mémoire et d'éviter les pannes de moteur dues à des out-of-memory exceptions.
- Sélecteur de lecture intelligent : cette implémentation améliore la latence de lecture en choisissant le segment de stockage optimal parmi les différents segments pour chaque lecture, ce qui améliore le débit de lecture. SysBench les tests ont montré une augmentation des performances allant jusqu'à 27 % pour les charges de travail d'écriture.

### Améliorations

- Résolution d'un problème où un réplica Aurora détecte un verrouillage partagé au démarrage du moteur.
- Résolution d'un problème possible sur un réplica Aurora lorsque le pointeur de vue de lecture sur le système de purge est NULL.

## Mises à jour du moteur de base de données Aurora MySQL : 26/10/2016 (version 1.8.1) (obsolète)

Version : 1.8.1

### Améliorations

- Correction d'un problème où les insertions en bloc utilisent des déclencheurs qui appellent les procédures AWS Lambda échouent.

- Correction d'un problème où la migration de catalogue échoue lorsque la validation automatique est désactivée de manière globale.
- Résolution d'un échec de connexion à Aurora lors de l'utilisation du protocole SSL et amélioration du groupe Diffie-Hellman pour faire face aux attaques. LogJam

### Intégration de correctifs de bogues MySQL.

- OpenSSL a modifié les paramètres de longueur de clé Diffie-Hellman en raison de ce problème. LogJam (Bogue n° 18367167)

## Mises à jour du moteur de base de données Aurora MySQL : 18/10/2016 (version 1.8) (obsolète)

Version : 1.8

### Nouvelles fonctionnalités

- AWS Lambda intégration — Vous pouvez désormais appeler de manière asynchrone une AWS Lambda fonction à partir d'un cluster de base de données Aurora à l'`mysql.lambda_async`aide de cette procédure. Pour plus d'informations, consultez Appel d'une fonction Lambda à partir d'un cluster de bases de données Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Chargement de données à partir d'Amazon S3 – Vous pouvez désormais charger du texte ou des fichiers XML d'un compartiment Amazon S3 dans votre cluster de bases de données Aurora à l'aide des commandes `LOAD DATA FROM S3` ou `LOAD XML FROM S3`. Pour plus d'informations, consultez Chargement de données dans un cluster de bases de données Amazon Aurora MySQL à partir de fichiers texte stockés dans un compartiment Amazon S3 dans le Guide de l'utilisateur Amazon Aurora.
- Migration du catalogue – Aurora conserve désormais les métadonnées de catalogue dans le volume de cluster pour prendre en charge la gestion des versions. Cela permet une migration de catalogue transparente entre différentes versions et restaurations.
- Maintenance et application de correctifs au niveau du cluster – Aurora gère désormais les mises à jour de maintenance pour un cluster de bases de données tout entier. Pour plus d'informations, consultez Entretien d'un cluster de base de données Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'un problème où un réplica Aurora se bloque s'il n'accorde pas de verrou de métadonnées dans une DDL en vol.
- Les réplicas Aurora sont autorisés à modifier des tables non-InnoDB pour faciliter la rotation des fichiers journaux CSV généraux et lents où `log_output=TABLE`.
- Correction d'un décalage dans la mise à jour des statistiques entre l'instance principale et un réplica Aurora. Sans ce correctif, les statistiques du réplica Aurora peuvent se trouver désynchronisées par rapport à celles de l'instance principale et engendrer un plan de requête différent (et sous-performant) sur un réplica Aurora.
- Correction d'une condition de concurrence qui garantit qu'un réplica Aurora n'acquiert pas de verrous.
- Correction d'un scénario rare où un réplica Aurora qui s'enregistre ou se désenregistre de l'instance principale peut échouer.
- Correction d'une condition de concurrence pouvant conduire à un blocage sur les instances `db.r3.large` lors de l'ouverture ou de la fermeture d'un volume.
- Correction d'un out-of-memory problème qui pouvait survenir en raison de la combinaison d'une charge de travail d'écriture importante et de défaillances du service de stockage distribué Aurora.
- Correction d'un problème de consommation élevée de l'UC en raison de la rotation du thread de purge en présence d'une transaction de longue durée.
- Correction d'un problème se produisant lorsqu'un schéma d'informations en cours d'exécution cherche à obtenir des informations sur les verrous dans des conditions de charge importante.
- Correction d'un problème relatif à un processus de diagnostic pouvant, dans de rares cas, entraîner des blocages et des redémarrages/basculements d'écritures Aurora dans des nœuds de stockage.
- Correction d'une condition où une table créée avec succès peut être supprimée pendant une récupération d'incident si l'incident s'est produit pendant le traitement d'une instruction `CREATE TABLE [if not exists]`.
- Correction d'un problème de rupture de la procédure de rotation des journaux lorsque le journal général et le journal lent ne sont pas stockés sur disque par atténuation des risques au catalogue.
- Correction d'un blocage se produisant lorsqu'un utilisateur crée une table temporaire au sein d'une fonction définie par l'utilisateur, puis utilise la fonction définie par l'utilisateur dans la liste de sélection de la requête.

- Correction d'un incident qui se produisait lors de la relecture des événements GTID. GTID n'est pas pris en charge par Aurora MySQL.

### Intégration de correctifs de bogues MySQL :

- Lors de l'abandon de tous les index sur une colonne à index multiples, InnoDB ne parvenait pas à bloquer une opération INDEX DEPOSER lorsqu'une contrainte de clé étrangère avait besoin d'un index. (Bogue n° 16896810)
- Résoudre l'incident lié à l'ajout d'une contrainte de clé étrangère. (Bogue n° 16413976)
- Correction d'un incident se produisant lors de la récupération d'un curseur dans une procédure stockée pendant l'analyse ou du vidage de la table. (Bogue n° 18158639)
- Correction d'un bogue d'incrémentation automatique se produisant lorsqu'un utilisateur modifie une table pour changer la valeur AUTO_INCREMENT à moins de la valeur maximale de la colonne d'incrémentation automatique. (Bogue n° 16310273)

## Mises à jour du moteur de base de données Aurora MySQL : 20/09/2016 (version 1.7.1) (obsolète)

Version : 1.7.1

### Améliorations

- Résout un problème dans le cadre duquel un réplica Aurora se bloque si le cache de recherche en texte intégral InnoDB est plein.
- Résout un problème dans le cadre duquel le moteur de base de données se bloque si un thread d'outil de traitement du pool de threads s'attend lui-même.
- Résout un problème dans le cadre duquel un réplica Aurora se bloque si un verrouillage de métadonnées dans une table est à l'origine d'un blocage.
- Résout un problème dans le cadre duquel le moteur de base de données se bloque en raison d'une condition de concurrence entre les deux threads d'outil de traitement dans le pool de threads.
- Résout un problème dans le cadre duquel un basculement inutile se produit dans des conditions de charge importante si l'agent de surveillance ne détecte pas la progression des opérations d'écriture dans le sous-système de stockage distribué.

# Mises à jour du moteur de base de données Aurora MySQL : 30/08/2016 (version 1.7.0) (obsolète)

Version : 1.7.0

## Nouvelles fonctions

- Planificateur NUMA – Le planificateur de tâches du moteur Aurora MySQL prend désormais en charge l'accès mémoire non uniforme (NUMA, Non-Uniform Memory Access). Cela réduit les conflits de sockets inter-CPU ce qui se traduit par des performances améliorées en termes de débit pour la classe d'instance de base de données `db.r3.8xlarge`.
- Fonctionnement de la lecture anticipée en parallèle de manière asynchrone en arrière-plan – La lecture anticipée en parallèle a été repensée pour améliorer les performances en utilisant un thread dédié pour réduire les conflits entre threads.
- Construction d'index améliorée (mode Lab) – La création d'index secondaires fonctionne désormais suivant un mode de construction de bas en haut, qui élimine les fractionnements de pages inutiles. Cela permet de réduire le temps de création d'un index ou de reconstruction d'une table. Cette fonction est désactivée par défaut et peut être activée en activant le mode Lab d'Aurora. Pour plus d'informations, consultez Mode Lab Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

## Améliorations

- Correction d'un problème de lenteur de la connexion en cas d'augmentation du nombre de connexions demandées pour une instance.
- Correction d'un problème où une défaillance générale se produisait si ALTER TABLE était exécutée sur une table partitionnée qui n'utilisait pas InnoDB.
- Correction d'un problème où une lourde charge de travail en écriture pouvait entraîner un basculement.
- Correction d'une assertion erronée qui entraînait une défaillance si RENAME TABLE était exécutée sur une table partitionnée.
- Stabilité améliorée en cas de restauration d'une transaction pendant une importante charge de travail d'insertion.
- Correction d'un problème où les index de recherche en texte intégral n'étaient pas fiables sur un réplica Aurora.

## Intégration de correctifs de bogues MySQL.

- Amélioration de l'évolutivité par partitionnement du verrou LOCK_grant. (Port WL #8355)
- L'ouverture du curseur sur SELECT dans la procédure stockée entraîne une erreur de segmentation. (Bogue de port n° 16499751)
- MySQL donne un faux résultat dans certains cas d'utilisation. (Bogue n° 11751794)
- Défaillance générale dans GET_SEL_ARG_FOR_KEYPART – causée par le correctif du bogue 11751794. (Bogue n° 16208709)
- Mauvais résultats pour une requête simple avec GROUP BY. (Bogue n° 17909656)
- Lignes supplémentaires sur requête semi-jointure avec prédicats de plage. (Bogue n° 16221623)
- L'ajout d'une clause ORDER BY à la suite d'une sous-requête IN pourrait entraîner le renvoi de lignes en double. (Bogue n° 16308085)
- Défaillance générale pour une requête avec balayage large pour GROUP BY, MySQL. (Bogue n ° 16222245)
- L'analyse d'index large avec prédicat d'entier à quota renvoie des données aléatoires. (Bogue n ° 16394084)
- Si l'optimiseur utilisait une analyse d'index large, le serveur pouvait quitter pendant la tentative de création d'une table temporaire. (Bogue n° 16436567)
- Alors qu'elle ne devrait pas le faire, la fonction COUNT(DISTINCT) comptait les valeurs NULL lorsque l'optimiseur utilisait l'analyse d'index large. (Bogue n° 17222452)
- Si une requête utilisait à la fois la fonction MIN()/MAX() et la fonction aggregate_function(DISTINCT) (par exemple, SUM(DISTINCT)) et était exécutée avec l'analyse d'index large, les valeurs produites par la fonction MIN()/MAX() étaient incorrectement définies. (Bogue n° 17217128)

# Mises à jour du moteur de base de données Aurora MySQL : 01/06/2016 (version 1.6.5) (obsolète)

Version : 1.6.5

## Nouvelles fonctions

- Stockage efficace des journaux binaires – Le stockage efficace des journaux binaires est maintenant activé par défaut pour tous les clusters de bases de données Aurora MySQL et n'est

pas configurable. Le stockage efficace des journaux binaires a été commercialisé avec la mise à jour d'avril 2016. Pour de plus amples informations, veuillez consulter Mises à jour du moteur de base de données Aurora MySQL : 06/04/2016 (version 1.6) (obsolète).

### Améliorations

- Stabilité améliorée pour les réplicas Aurora lorsque l'instance principale rencontre une lourde charge de travail.
- Stabilité améliorée pour les réplicas Aurora lorsque vous exécutez des requêtes sur les tables partitionnés et les tables dont le nom comporte des caractères spéciaux.
- Correction des problèmes de connexion en cas d'utilisation de connexion sécurisée.

### Intégration de correctifs de bogues MySQL.

- SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)

## Mises à jour du moteur de base de données Aurora MySQL : 06/04/2016 (version 1.6) (obsolète)

Version : 1.6

Cette mise à jour inclut les améliorations suivantes :

### Nouvelles fonctions

- Lecture anticipée parallèle – La lecture anticipée parallèle est désormais activée par défaut pour tous les clusters de bases de données Aurora MySQL et n'est pas configurable. La lecture anticipée parallèle a été commercialisée avec la mise à jour de décembre 2015. Pour plus d'informations, consultez Mises à jour du moteur de base de données Aurora MySQL : 03/12/2015 (version 1.4) (obsolète).

  En plus de permettre la lecture anticipée parallèle par défaut, cette version inclut les améliorations suivantes à la lecture anticipée parallèle :

  - Logique améliorée pour que la lecture anticipée parallèle soit moins agressive, ce qui est utile lorsque votre cluster DB rencontre de nombreuses charges de travail parallèles.

  - Stabilité améliorée des tables plus petites.
- Stockage efficace des journaux binaires (mode Lab) – Les fichiers journaux binaires MySQL sont désormais stockés plus efficacement dans Aurora MySQL. La nouvelle implémentation de stockage permet de supprimer les fichiers journaux binaires beaucoup plus tôt et améliore les performances système d'une instance dans un cluster de bases de données Aurora MySQL qui est un maître de réplication de journal binaire.

  Pour activer le stockage efficace des journaux binaires, affectez au paramètre `aurora_lab_mode` la valeur `1` dans le groupe de paramètres pour votre instance principale ou votre réplica Aurora. Le paramètre `aurora_lab_mode` est un paramètre au niveau d'une instance qui figure par défaut dans le groupe de paramètres `default.aurora5.6`. Pour plus d'informations sur la modification d'un groupe de paramètres de base de données, consultez Modification de paramètres dans un groupe de paramètres de bases de données dans le Guide de l'utilisateur Amazon Aurora. Pour plus d'informations sur les groupes de paramètres et Aurora MySQL, consultez Paramètres de configuration d'Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

  Activez uniquement le stockage efficace des journaux binaires pour des instances dans un cluster de bases de données Aurora MySQL qui sont des instances du maître de réplication de journaux binaires MySQL.
- Variable système AURORA_VERSION – Vous pouvez maintenant obtenir la version Aurora de votre cluster de bases de données Aurora MySQL en recherchant la variable système `AURORA_VERSION`.

  Pour obtenir la version d'Aurora, utilisez l'une des requêtes suivantes :

  ```
  select AURORA_VERSION();
  select @@aurora_version;
  show variables like '%version';
  ```

  Vous pouvez également voir la version d'Aurora AWS Management Console lorsque vous modifiez un cluster de base de données, ou en appelant la describe-db-engine-versions AWS CLI commande ou l'opération d'API Describe DBEngine Versions.
- Métrique d'utilisation de la mémoire du gestionnaire de verrous – Les informations sur l'utilisation de la mémoire du gestionnaire de verrous sont désormais disponibles sous la forme d'une métrique.

Pour obtenir la métrique d'utilisation de la mémoire du gestionnaire de verrous, utilisez l'une des requêtes suivantes :

```
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');
```

## Améliorations

- Stabilité améliorée pendant la récupération des journaux binaires et de transaction XA.
- Correction d'un problème de mémoire résultant d'un grand nombre de connexions.
- Amélioration de la précision dans les métriques suivantes : `Read Throughput`, `Read IOPS`, `Read Latency`, `Write Throughput`, `Write IOPS`, `Write Latency` et `Disk Queue Depth`.
- Correction d'un problème de stabilité entraînant un démarrage lent pour des instances importantes après un incident.
- Amélioration de la concurrence dans le dictionnaire de données concernant les mécanismes de synchronisation et l'expulsion du cache.
- Améliorations de la stabilité et des performances pour les réplicas Aurora :
  - Correction d'un problème de stabilité pour les réplicas Aurora lors de charges de travail en écriture intensive ou à forte densité pour l'instance principale.
  - Amélioration du retard du réplica pour les instances db.r3.4xlarge et db.r3.8xlarge.
  - Amélioration des performances en réduisant les conflits entre les applications d'enregistrements de journal et les lectures simultanées sur un réplica Aurora.
  - Correction d'un problème d'actualisation des statistiques concernant les réplicas Aurora pour les statistiques nouvellement créés ou mises à jour.
  - Amélioration de la stabilité pour les réplicas Aurora lorsqu'il existe de nombreuses transactions sur l'instance principale et des lectures simultanées sur les réplicas Aurora sur les mêmes données.
  - Amélioration de la stabilité pour les réplicas Aurora lorsque de l'exécution des instructions `UPDATE` et `DELETE` avec des instructions `JOIN`.
  - Amélioration la stabilité des réplicas Aurora lors de l'exécution d'instructions `INSERT ... SELECT`.

### Intégration de correctifs de bogues MySQL.

- BACKPORT Bug #18694052 FIX FOR ASSERTION `!M_ORDERED_REC_BUFFER' FAILED TO 5.6 (Port Bug #18305270)
- SEGV IN MEMCPY(), HA_PARTITION::POSITION (Port Bug # 18383840)
- WRONG RESULTS WITH PARTITIONING,INDEX_MERGE AND NO PK (Port Bug # 18167648)
- FLUSH TABLES FOR EXPORT: ASSERTION IN HA_PARTITION::EXTRA (Port Bug # 16943907)
- SERVER CRASH IN VIRTUAL HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST (Port Bug # 16164031)
- RANGE OPTIMIZER CRASHES IN SEL_ARG::RB_INSERT() (Port Bug # 16241773)

## Mises à jour du moteur de base de données Aurora MySQL : 11/01/2016 (version 1.5) (obsolète)

Version : 1.5

Cette mise à jour inclut les améliorations suivantes :

### Améliorations

- Correction d'une pause de 10 s des opérations d'écriture pour les instances inactives au cours des déploiements du stockage Aurora.
- La lecture anticipée logique fonctionne à présent quand `innodb_file_per_table` a la valeur No. Pour plus d'informations sur la lecture anticipée logique, consultez Mises à jour du moteur de base de données Aurora MySQL : 03/12/2015 (version 1.4) (obsolète).
- Correction de problèmes liés à la reconnexion des réplicas Aurora à l'instance principale. Cette amélioration corrige également un problème qui se posait lorsque vous spécifiiez une valeur importante pour le paramètre `quantity` dans le cadre du test des échecs des réplicas Aurora à l'aide de requêtes d'injection d'erreurs. Pour plus d'informations, consultez Test d'une défaillance d'un réplica Aurora dans le Guide de l'utilisateur Amazon Aurora.
- Amélioration de la supervision des retards et des redémarrages des réplicas Aurora.
- Correction d'un problème qui entraînait le retard, la désinscription puis le redémarrage d'un réplica Aurora.

- Correction d'un problème qui se posait lors de l'exécution de la commande `show innodb status` au cours d'un interblocage.
- Correction d'un problème de basculement pour les instances volumineuses lors d'un débit élevé d'écriture.

### Intégration de correctifs de bogues MySQL.

- Traitement d'un correctif incomplet dans la recherche en texte intégral MySQL affectant les tables dans lesquelles le nom de base de données commence par un chiffre. (Bogue de port n ° 17607956)

## Mises à jour du moteur de base de données Aurora MySQL : 03/12/2015 (version 1.4) (obsolète)

Version : 1.4

Cette mise à jour inclut les améliorations suivantes :

### Nouvelles fonctions

- Insertion rapide – Accélère les insertions parallèles triées par leur clé primaire. Pour plus d'informations, consultez Améliorations des performances Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.
- Performances de lecture des grands ensembles de données – Aurora MySQL détecte automatiquement une lourde charge de travail d'I/O et démarre plus de threads afin d'améliorer les performances du cluster de bases de données. Le planificateur Aurora examine l'activité d'I/O et décide d'ajuster dynamiquement le nombre optimal de threads dans le système en l'adaptant rapidement en fonction des charges de travail exigeantes en I/O et exigeantes en UC, sans frais généraux importants.
- Lecture anticipée parallèle – Améliore les performances des analyses de l'arbre B (B-Tree) qui sont trop grandes pour la mémoire disponible sur votre instance principale ou votre réplica Aurora (y compris les requêtes de plage). La lecture anticipée parallèle détecte automatiquement les modèles de lecture de page et récupère au préalable les pages dans le cache des tampons pour anticiper le moment où elles seront requises. La lecture anticipée parallèle traite plusieurs tables au même moment, dans la même transaction.

## Améliorations :

- Correction des problèmes de brève disponibilité de base de données Aurora au cours des déploiements du stockage Aurora.
- Application correcte de la limite `max_connection`.
- Amélioration de la purge des fichiers journaux binaires quand Aurora est l'entité principale de journalisation et que la base de données redémarre après une lourde charge de données.
- Correction de problèmes de gestion de mémoire avec le cache des tables.
- Ajout de la prise en charge des très grandes pages dans le cache des tampons de mémoire partagée pour accélérer la récupération.
- Correction d'un problème lié à non-initialisation du stockage local des threads.
- Autorisation de 16 000 connexions par défaut.
- Pool de threads dynamiques pour les charges de travail exigeantes en I/O.
- Correction d'un problème d'invalidation correcte des vues impliquant UNION dans le cache de requêtes.
- Correction d'un problème de stabilité avec le thread stats du dictionnaire.
- Correction d'une fuite de mémoire dans le sous-système du dictionnaire associée à l'expulsion du cache.
- Correction d'un problème de latence de lecture élevée sur les réplicas Aurora lorsque la charge d'écriture est très faible sur le maître.
- Correction de problèmes de stabilité sur les réplicas Aurora lors de la réalisation d'opérations sur les tables partitionnées DDL telles que ALTER TABLE... REORGANIZE PARTITION sur le maître.
- Correction de problèmes de stabilité sur les réplicas Aurora lors d'une croissance du volume.
- Correction d'un problème de performance pour les analyses sur les index non organisés en cluster sur les réplicas Aurora.
- Correction d'un problème de stabilité entraînant un retard des réplicas Aurora et leur désinscription et redémarrage éventuels.

## Intégration de correctifs de bogues MySQL.

- SEGV dans FTSPARSE(). (Bogue n° 16446108)
- Le dictionnaire de données InnoDB n'est pas mis à jour lorsque la colonne est renommée. (Bogue n° 19465984)

- Incident FTS après qu'une table a été renommée dans une autre base de données. (Bogue n ° 16834860)
- L'échec de la préparation du déclenchement sur les tables tronquées entraîne l'erreur 1054. (Bogue n° 18596756)
- Des modifications des métadonnées peuvent entraîner des problèmes de déclenchement d'exécution. (Bogue n° 18684393)
- La matérialisation n'est pas choisie pour le champ UTF8 VARCHAR long. (Bogue n° 17566396)
- Plan d'exécution médiocre pour ORDER BY avec la limite X. (Bogue n°16697792)
- Bogue de rétroportage n° 11765744 pour 5.1, 5.5 et 5.6. (Bogue n° 17083851)
- Problème de mutex dans SQL/SQL_SHOW.CC entraînant. SIG6 Source probable FILL_VARIABLES. (Bogue n° 20788853)
- Bogue de rétroportage n° 18008907 pour les versions 5.5+. (Bogue n° 18903155)
- Adaptation d'un correctif pour une erreur de dépassement de capacité de pile dans MySQL 5.7. (Bogue n° 19678930)

# Mises à jour du moteur de base de données Aurora MySQL : 16/10/2015 (versions 1.2, 1.3) (obsolète)

Versions : 1.2, 1.3

Cette mise à jour inclut les améliorations suivantes :

## Correctifs

- out-of-memoryProblème résolu dans le nouveau gestionnaire de verrous avec des transactions de longue durée
- Résolution d'une faille de sécurité lors de la réplication avec des bases de données MySQL non-RDS
- Mise à jour pour garantir que les écritures de quorum sont réessayées correctement en cas de défaillances de stockage
- Mise à jour pour signaler plus précisément un retard du réplica
- Amélioration de la performance en réduisant les conflits lorsque de nombreuses transactions simultanées essaient de modifier une même ligne

- Résolution de l'invalidation du cache de requêtes pour les vues qui sont créées en joignant deux tables
- Désactivation du cache de requêtes pour les transactions avec isolation `UNCOMMITTED_READ`

## Améliorations

- Meilleures performances pour les requêtes lentes de catalogue sur les caches à chaud
- Amélioration de la simultanéité dans les statistiques de dictionnaire
- Meilleure stabilité pour le nouveau gestionnaire de ressources du cache de requêtes, la gestion de l'extension, les fichiers stockés dans le stockage intelligent Amazon Aurora et les écritures par lots des enregistrements de journaux

## Intégration de correctifs de bogues MySQL.

- L'arrêt d'une requête dans innodb entraîne finalement un incident avec une assertion. (Bogue n° 1608883)
- Pour un échec lors de la création d'un nouveau thread pour le planificateur d'événement, l'exécution d'un événement ou une nouvelle connexion, aucun message n'a été consigné dans le journal des erreurs. (Bogue n° 16865959)
- Si une connexion a changé la base de données par défaut alors que simultanément une autre connexion a exécuté SHOW PROCESSLIST, la seconde connexion a pu accéder à de la mémoire non valide lors de la tentative d'affichage de la mémoire de base de données par défaut de la première connexion. (Bogue n° 11765252)
- PURGE BINARY LOGS, par conception, ne supprime pas les fichiers journaux binaires qui sont utilisés ou actifs, mais n'a pas fourni aucun avis lorsque cela s'est produit. (Bogue n° 13727933)
- Pour certaines instructions, des fuites de mémoire pouvaient être générées lorsque l'optimiseur supprimait des clauses de sous-requête superflues. (Bogue n° 15875919)
- Pendant l'arrêt, le serveur pouvait essayer de verrouiller un mutex non initialisé. (Bogue n° 16016493)
- Une instruction préparée qui utilisait GROUP_CONCAT() et une clause ORDER BY qui nommait plusieurs colonnes pouvait entraîner l'arrêt du serveur. (Bogue n° 16075310)
- L'instrumentation du schéma de performance était manquante pour les threads de travailleur de réplicas. (Bogue n° 16083949)

- `STOP SLAVE` pouvait provoquer un blocage lorsqu'il était émis en même temps qu'une instruction telle que SHOW STATUS ayant extrait les valeurs pour une ou plusieurs des variables d'état `Slave_retried_transactions`, `Slave_heartbeat_period`, `Slave_received_heartbeats`, `Slave_last_heartbeat` ou `Slave_running`. (Bogue n ° 16088188)
- Une requête en texte intégral utilisant le mode booléen pouvait ne retourner aucun résultat dans certains cas où le terme recherché était une phrase entre guillemets. (Bogue n° 16206253)
- La tentative de l'optimiseur de supprimer les clauses de sous-requête redondantes générait une assertion lors de l'exécution d'une instruction préparée avec une sous-requête dans la clause ON d'une jointure dans une sous-requête. (Bogue n° 16318585)
- GROUP_CONCAT instable, incident dans ITEM_SUM::CLEAN_UP_AFTER_REMOVAL. (Bogue n ° 16347450)
- Une tentative de remplacement de la liste de mots vides de recherche en texte intégral InnoDB par défaut en créant une table InnoDB avec la même structure que INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD générait une erreur. (Bogue n ° 16373868)
- Après que le thread client sur un travailleur a effectué une opération FLUSH TABLES WITH READ LOCK suivie de quelques mises à jour sur le maître, le travailleur restait bloqué lors de l'exécution de `SHOW SLAVE STATUS`. (Bogue n° 16387720)
- Lors de l'analyse d'une chaîne de recherche délimitée telle que « abc-def » dans une recherche en texte intégral, InnoDB utilise à présent les mêmes délimiteurs de mot que MyISAM. (Bogue n ° 16419661)
- Incident dans FTS_AST_TERM_SET_WILDCARD. (Bogue n° 16429306)
- SEGFAULT dans FTS_AST_VISIT() pour le test RQG de recherche en texte intégral. (Bogue n ° 16435855)
- Pour les versions de débogage, lorsque l'optimiseur supprimait un Item_ref pointant sur une sous-requête, il entraînait un arrêt du serveur. (Bogue n° 16509874)
- La recherche en texte intégral sur des tables InnoDB échouait lors de recherches d'expressions littérales combinées aux opérateurs + ou -. (Bogue n° 16516193)
- `START SLAVE`a échoué lorsque le serveur a été démarré avec les options -- master-info-repository =TABLE relay-log-info-repository =TABLE et avec autocommit défini sur 0, ainsi que. `--skip-slave-start` (Bogue n° 16533802)
- Les résultats d'une recherche en texte intégral InnoDB très étendue pouvaient consommer une quantité excessive de mémoire. (Bogue n° 16625973)

- Dans les versions de débogage, une assertion pouvait survenir dans OPT_CHECK_ORDER_BY lors de l'utilisation de données binaires directement dans une chaîne de recherche, car les données binaires peuvent inclure des octets NULL et d'autres caractères non significatifs. (Bogue n ° 16766016)
- Pour certaines instructions, des fuites de mémoire pouvaient être générées lorsque l'optimiseur supprimait des clauses de sous-requête superflues. (Bogue n° 16807641)
- Il était possible de générer un interblocage après l'émission de FLUSH TABLES WITH READ LOCK en émettant `STOP SLAVE` dans une nouvelle connexion à l'esclave, puis en émettant `SHOW SLAVE STATUS` à l'aide de la connexion d'origine. (Bogue n° 16856735)
- GROUP_CONCAT() avec un séparateur non valide pouvait causer un arrêt du serveur. (Bogue n ° 16870783)
- Le serveur a effectué un verrouillage excessif sur les mutex LOCK_active_mi et active_mi->rli->data_lock pour toute instruction SHOW STATUS LIKE 'modèle', même lorsque le modèle ne correspond pas aux variables d'état qui utilisent ces mutex (`Slave_heartbeat_period`, `Slave_last_heartbeat`, `Slave_received_heartbeats`, `Slave_retried_transactions`, `Slave_running`). (Bogue n° 16904035)
- Une recherche en texte intégral utilisant le modificateur IN BOOLEAN MODE entraînait un échec d'assertion. (Bogue n° 16927092)
- Une recherche en texte intégral sur les tables InnoDB échouait pour les recherches qui utilisaient l'opérateur booléen +. (Bogue n° 17280122)
- Interblocage quadridirectionnel : zombies, purge des journaux binaires, show processlist, show binlogs. (Bogue n° 17283409)
- Lorsqu'un thread SQL qui attendait un verrou de validation était supprimé et redémarré, une transaction était ignorée sur le travailleur. (Bogue n° 17450876)
- Un échec de recherche en texte intégral InnoDB survenait en raison d'un jeton « non terminé ». La chaîne et la longueur de chaîne devaient être transmises pour la comparaison de chaîne. (Bogue n ° 17659310)
- Un grand nombre de tables InnoDB partitionnées pouvaient consommer beaucoup plus de mémoire lorsqu'elles étaient utilisées dans MySQL 5.6 ou 5.7 que la mémoire utilisée par les mêmes tables dans les versions précédentes du serveur MySQL. (Bogue n° 17780517)
- Pour les requêtes en texte intégral, ne pas vérifier si num_token était inférieur à max_proximity_item pouvait entraîner une assertion. (Bogue n° 18233051)

- Certaines requêtes relatives aux tables INFORMATION_SCHEMA TABLES et COLUMNS pouvaient entraîner une utilisation excessive de mémoire lorsque de nombreuses tables InnoDB étaient vides. (Bogue n° 18592390)
- Lors de la validation d'une transaction, un indicateur est désormais utilisé pour vérifier si un thread a été créé, plutôt que la vérification du thread lui-même, qui utilise plus de ressources, en particulier lors de l'exécution du serveur avec master_info_repository=TABLE. (Bogue n° 18684222)
- Si un thread client sur un travailleur exécutait FLUSH TABLES WITH READ LOCK alors que le maître exécutait une commande DML, l'exécution de `SHOW SLAVE STATUS` dans le même client se bloquait, entraînant un interblocage. (Bogue n° 19843808)
- Le tri selon un résultat GROUP_CONCAT() pouvait causer l'arrêt du serveur. (Bogue n° 19880368)

# Mises à jour du moteur de base de données Aurora MySQL : 24/08/2015 (version 1.1) (obsolète)

Version : 1.1

Cette mise à jour inclut les améliorations suivantes :

- Améliorations de la stabilité de réplication lors de la réplication avec une base de données MySQL (réplication des journaux binaires). Pour plus d'informations sur la réplication d'Aurora MySQL avec MySQL, consultez Réplication avec Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora.
- Limite de 1 gigaoctet (Go) sur la taille des journaux relais accumulés pour un cluster de bases de données Aurora MySQL qui est un travailleur de réplication. Ceci améliore la gestion des fichiers pour les clusters DB Aurora.
- Améliorations de la stabilité dans les zones de lecture anticipée, relations de clé étrangère récursives et réplication Aurora.
- Intégration de correctifs de bogues MySQL.
  - Les bases de données InnoDB dont les noms commencent par un chiffre entraînent une erreur d'analyse de la recherche de texte intégral (FTS). (Bogue n° 17607956)
  - Les recherches de texte intégral InnoDB échouent dans les bases de données dont les noms commencent par un chiffre. (Bogue n° 17161372)
  - Pour les bases de données InnoDB sous Windows, l'ID d'objet de la recherche de texte intégral (FTS) n'est pas au format hexadécimal attendu. (Bogue n° 16559254)

  - Une régression de code présentée dans MySQL 5.6 a impacté de manière négative les performances DROP TABLE et ALTER TABLE. Cela pourrait entraîner une baisse des performances entre MySQL Server 5.5.x et 5.6.x. (Bogue n° 16864741)
- Journalisation simplifiée pour réduire la taille des fichiers journaux et le volume de stockage qu'ils requièrent.

# Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL

Les sections suivantes identifient les bogues MySQL qui ont été corrigés par les mises à jour du moteur de base de données Aurora MySQL.

Rubriques


- Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x
- Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x
- Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 1.x


## Corrections de bogues effectuées par les mises à jour du moteur de base de données d'Aurora MySQL 3.x

La version d'Aurora compatible avec MySQL 8.0 contient toutes les corrections de bogues MySQL jusqu'à la version compatible avec MySQL correspondante. Le tableau suivant identifie les bogues MySQL supplémentaires qui ont été corrigés par les mises à jour du moteur de base de données Aurora MySQL, ainsi que la mise à jour dans laquelle ils ont été corrigés.

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora | 8,0,39 | 3,08,0 | • Correction d'un problème qui provoquait l'omission incorrecte de `NULL` valeurs dans le jeu de résultats pour certaines requêtes comportant `JOIN` les deux `UNION` opérations. (Correctif de bogue communautaire #114301) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
| --- | --- | --- | --- |
| MySQL 2024-11-18 (version 3.08.0, compatible avec MySQL 8.0.39) | | | |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL 2024-06-04 (version 3.07.0, compatible avec MySQL 8.0.36) | 8,0,36 | 3,07,0 | • Correction d'un problème en raison duquel la valeur de la ligne de cache pouvait être calculée de manière incorrecte, ce qui provoquait un échec lors du redémarrage de la base de données sur les instances basées sur Graviton. (Correctif de bogue communautaire #35479763)<br>• Correction d'un problème en raison duquel certaines instances de sous-requêtes dans les routines stockées n'étaient pas toujours traitées correctement. (Correctif de bogue communautaire #35377192)<br>• Correction d'un problème qui pouvait entraîner une augmentation de l'utilisation du processeur en raison de la rotation des certificats TLS en arrière-plan (Community Bug Fix #34284186).<br>• Correction d'un problème en raison duquel InnoDB autorisait l'ajout de `INSTANT` colonnes aux tables dans le schéma |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| | | | système MySQL dans les versions d'Aurora MySQL inférieures à 3.05, ce qui pouvait entraîner la fermeture inattendue du serveur (redémarrage de l'instance de base de données) après la mise à niveau vers la version 3.05.0 d'Aurora MySQL. (Correctif de bogue communautaire #35625510). |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL 2024-03-07 (version 3.06.0, compatible avec MySQL 8.0.34) | 8,0,34 | 3,06,0 | • Correction d'un problème en raison duquel la valeur de la ligne de cache pouvait être calculée de manière incorrecte, ce qui provoquait un échec lors du redémarrage de la base de données sur une instance de Graviton. (Correctif de bogue communautaire #35479763)<br>• Correction d'un problème en raison duquel certaines instances de sous-requêtes dans les routines stockées n'étaient pas toujours traitées correctement. (Correctif de bogue communautaire #35377192)<br>• Correction d'un problème qui pouvait entraîner une augmentation de l'utilisation du processeur en raison de la rotation des certificats TLS en arrière-plan. (Correctif du bogue #34284186)<br>• Correction d'un problème en raison duquel InnoDB autorisait l'ajout de `INSTANT` colonnes aux tables dans le schéma système MySQL dans les versions d'Aurora MySQL |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| | | | inférieures à 3.05, ce qui pouvait entraîner la fermeture inattendue du serveur (redémarrage de l'instance de base de données) après la mise à niveau vers la version 3.05.0 d'Aurora MySQL. (Correctif de bogue Community n° 35625510) |
| Mises à jour du moteur de base de données Aurora MySQL 2024-01-31 (version 3.05.2, compatible avec MySQL 8.0.32) | 8,0,32 | 3,05.2 | • Correction d'un problème en raison duquel un nombre excessif de lectures de disque étaient `records_in_range` effectuées pour les `INSERT` opérations, ce qui entraînait une baisse progressive des performances. (Correctif de bogue communautaire #34976138) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL 21/11/2023 (version 3.05.1, compatible avec MySQL 8.0.32) | 8,0,32 | 3,05.1 | • Correction d'un problème dans InnoDB : si une `INSTANT ADD` colonne était ajoutée à une table MySQL dans un schéma système entre les versions 3.01 à Aurora MySQL 3.04 et après la mise à niveau d'Aurora MySQL vers la version 3.05.0, le serveur se fermait de manière inattendue DMLs sur ces tables. (Correctif de bogue Community n° 35625510) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 25/10/2023 (version 3.0 compatible avec MySQL 8.0 | 8,0,32 | 3,05,0 | • Correction d'un problème pouvant entraîner une utilisation accrue du processeur en raison de la rotation des certificats TLS en arrière-plan (correctif du bogue n° 34284186) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL 2024-03-15 (version 3.04.2, compatible avec MySQL 8.0.28) | 8,0,28 | 3.04.2 | • Correction d'un problème en raison duquel la valeur de la ligne de cache pouvait être calculée de manière incorrecte, ce qui provoquait un échec lors du redémarrage de la base de données sur les instances basées sur Graviton. (Correctif de bogue communautaire #35479763)<br>• L'exécution répétée d'une routine stockée, ayant comme sous-requête une instruction SELECT contenant plusieurs ou XOR conditions AND OR, a entraîné une consommation excessive et éventuellement un épuisement de la mémoire virtuelle. (Correctif de bogue communautaire #33852530) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL 13/11/2023 (version 3.04.1, compatible avec MySQL 8.0.28) | 8,0,28 | 3.04.1 | • Correction d'un problème pouvant entraîner une utilisation accrue du processeur en raison de la rotation des certificats TLS en arrière-plan (correctif du bogue n° 34284186) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL 31/07/2023 (version 3.04.0, compatible avec MySQL 8.0.28) | 8,0,28 | 3,04,0 | • Correction d'un problème de déplacement d'un bloc de mémoire tampon contenant une page de table temporaire intrinsèque lors du parcours d'une page, provoquant un échec d'assertion (Bogue n° 33715694)<br>• InnoDB : empêcher les opérations DDL en ligne d'accéder à la out-of-bounds mémoire (bogue n° 34750489, bogue n° 108925)<br>• Correction d'un problème qui pouvait parfois produire des résultats de requête incorrects lors du traitement d'instructions SQL complexes composées de plusieurs expressions de table communes imbriquées (CTEs) (bogue n° 34572040, bogue n° 34634469, bogue n° 33856374) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL 08/2023 (version 3.03.3) (obsolète) | 8,0,26 | 3.03.3 | • Correction d'un problème pouvant entraîner une utilisation accrue du processeur en raison de la rotation des certificats TLS en arrière-plan (correctif du bogue n° 34284186) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
| --- | --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL 29/08/2023 (version 3.03.2) (obsolète) | 8,0,26 | 3.03.2 | • Correction d'un problème qui pouvait parfois produire des résultats de requête incorrects lors du traitement d'instructions SQL complexes composées de plusieurs expressions de table communes imbriquées (CTEs) (Bug #34572040, Bug #34634469, Bug #33856374)<br>• InnoDB : échec d'assertion provoqué par une condition de concurrence entre des threads qui tentaient de désinitialiser et d'initialiser des statistiques pour une même table (bogue n ° 33135425).<br>• InnoDB : Empêcher les opérations DDL en ligne d'accéder à la out-of-bounds mémoire (Bug #34750489, Bug #108925) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
| --- | --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL 11/05/2023 (version 3.03.1) (obsolète) | 8,0,26 | 3.03.1 | • Correction d'un problème de relocalisation d'un bloc de mémoire tampon contenant une page de table temporaire intrinsèque lors du parcours d'une page, provoquant un échec d'assertion (bogue n ° 33715694) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL : 01/01 (version 3.03.0) (obsolète) | 8,0,26 | 3,03,0 | • Correction d'un problème lors duquel certains types de colonnes, dont JSON et TEXT, épuisaient parfois la mémoire tampon de tri si sa taille n'était pas au moins 15 fois supérieur e à celle de la plus grande ligne du tri. Désormais, le tampon de tri ne doit être que 15 fois supérieur à la plus grande clé de tri. (Bogues n° 103325, 105532, 32738705 et 33501541)<br>• Correction d'un problème lors duquel InnoDB ne gérait pas toujours correctement certains noms légaux de partitions de table. (Bogue n° 32208630)<br>• Correction d'un problème susceptible, dans certaines conditions, de renvoyer des résultats incorrects en raison d'un calcul inexact de la propriété de nullabilité lors de l'exécution d'une requête avec une condition OR. (Bogue n ° 34060289)<br>• Correction d'un problème susceptible, dans certaines conditions, de renvoyer des |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| | | | résultats incorrects lorsque les deux conditions suivantes sont remplies :<br>• une table dérivée est fusionnée dans le bloc de requête externe.<br>• la requête comprend une jointure gauche et une sous-requête IN.<br><br>(Bogue n° 34060289)<br>• Correction d'un problème de génération de valeurs incorrect es lorsque la valeur maximale de la colonne de type entier était dépassée. L'erreur était due à l'absence de prise en compte de la valeur maximale de colonne. La précédente valeur AUTO_INCREMENT valide aurait dû être renvoyée dans ce cas, ce qui a provoqué un doublon de clé. (Bogues n ° 87926 et 26906787)<br>• Correction d'un problème lors duquel il n'était pas possible de révoquer le privilège DROP sur le schéma de performance. (Bogue n° 33578113) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| | | | • Correction d'un problème lors duquel une procédure stockée contenant une instruction IF utilisant EXISTS, qui agissait sur une ou plusieurs tables supprimées et recréées entre les exécutions, ne s'exécutait pas correctement lors des invocations suivantes après la première. (Bogue n° 32855634).<br>• Correction d'un problème lors duquel une requête référence une vue dans une sous-requête et un bloc de requête externe peut provoquer un redémarrage inattendu. (Bogue n° 32324234) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL le 18 novembre 2018 (version 3.02.2) (obsolète) | 8,0,23 | 3.02.2 | • Correction d'un problème susceptible, dans certaines conditions, de renvoyer des résultats incorrects en raison d'un calcul inexact de la propriété de nullabilité lors de l'exécution d'une requête avec une condition OR. (Bogue n ° 34060289)<br>• Correction d'un problème susceptible, dans certaines conditions, de renvoyer des résultats incorrects lorsque les deux conditions suivantes sont remplies :<br>  • Une table dérivée est fusionnée dans le bloc de requête externe.<br>  • La requête inclut une jointure gauche et une sous-requête IN. (Bogue n° 34060289)<br>• Correction d'un problème de révocation du privilège DROP sur le schéma de performance. (Bogue n° 33578113)<br>• Correction d'un problème lors duquel une procédure stockée contenant une instruction IF |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| | | | utilisant EXISTS, qui agissait sur une ou plusieurs tables supprimées et recréées entre les exécutions, ne s'exécutait pas correctement lors des invocations suivantes après la première. (Bogue MySQL n° 32855634).<br>• Correction d'un problème de génération de valeurs AUTO_INCREMENT incorrectes lorsque la valeur maximale de la colonne de type entier était dépassée. L'erreur était due à l'absence de prise en compte de la valeur maximale de colonne. La précédente valeur AUTO_INCREMENT valide aurait dû être renvoyée dans ce cas, ce qui a provoqué un doublon de clé. (Bogues n° 87926 et 26906787)<br>• Correction d'un problème qui pouvait entraîner un échec lors de la mise à niveau d'un cluster de base de données Aurora MySQL version 1 (compatible avec MySQL 5.6) contenant une table créée par l'utilisateur avec une certaine table IDs. |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| | | | L'attribution de ces tables IDs peut entraîner un conflit entre les tables du dictionnaire de données IDs lors de la mise à niveau d'Aurora MySQL version 2 (compatible avec MySQL 5.7) vers Aurora MySQL version 3 (compatible avec MySQL 8.0). (Bogue n° 33919635) |
| Mises à jour du moteur de base de données Aurora MySQL 20/04/2020 (version 3.02.0) (obsolète) | 8,0,23 | 3,02,0 | Correction d'une mauvaise gestion des tables temporaires utilisées pour les curseurs dans des procédures stockées qui pouvait entraîner un comportement inattendu du serveur. (Bogue n° 32416811) |

| Mise à jour du moteur de base de données | Version compatible avec MySQL | Version | Bogues MySQL corrigés |
|---|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL 15/04/15 (version 3.01.1) (obsolète) | 8,0,23 | 3.01.1 | Correction d'une mauvaise gestion des tables temporaires utilisées pour les curseurs dans des procédures stockées qui pouvait entraîner un comportem ent inattendu du serveur. (Bogue n ° 32416811) |

## Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 2.x

La version compatible avec MySQL 5.7 Aurora contient toutes les corrections de bogues MySQL jusqu'à MySQL 5.7.44. Le tableau suivant identifie les bogues MySQL supplémentaires qui ont été corrigés par les mises à jour du moteur de base de données Aurora MySQL, ainsi que la mise à jour dans laquelle ils ont été corrigés.

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de | 2.12.3 | • Correction d'un problème en raison duquel les tables temporaires liées à des déclencheurs lors de l'exécution |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| données Aurora MySQL 2024-07-09 (version 2.12.3, compatible avec MySQL 5.7.44) - Version RDS Extended Support | | d'instructions pouvaient provoquer un redémarrage inattendu du moteur de base de données.<br>• Correction d'un défaut qui pouvait entraîner la fermeture du serveur lorsque des instructions à table unique `UPDATE` et `DELETE` des instructions utilisant des expressions indexées étaient exécutées en tant qu'instructions préparées. (Insecte #29257254) |
| Mises à jour du moteur de base de données Aurora MySQL 28/12/2023 (version 2.12.1, compatible avec MySQL 5.7.40) - Version RDS Extended Support | 2.12.1 | • Correction d'un problème qui pouvait entraîner le blocage des connexions à distance existantes et nouvelles lorsqu'el les étaient exécutées simultanément avec l'instruction `SHOW PROCESSLIST` (bogue Community n° 34857411)<br>• Réplication : certains événements de journal binaire n'étaient pas toujours gérés correctement (bogue n° 34617506).<br>• Correction du traitement des jetons à caractère unique par un plug-in d'analyse de recherche en texte intégral (bogue n ° 35432973) |
| Mises à jour du moteur de base de données Aurora MySQL 25/07/2023 (version 2.12.0, compatible avec MySQL 5.7.40) - Version RDS Extended Support | 2.12.0 | • Correction d'un problème susceptible d'entraîner une utilisati on plus élevée du processeur en raison de la rotation des certificats TLS en arrière-plan. (Correctif du bogue #34284186 ) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL 17/10/2023 (version 2.11.4, compatible avec MySQL 5.7.12) - Version RDS Extended Support | 2.11.4 | • Réplication : certains événements du journal binaire n'étaient pas toujours gérés correctement. (Bogue n° 34617506)<br>• Correction d'un problème susceptible d'entraîner une utilisation plus élevée du processeur en raison de la rotation des certificats TLS en arrière-plan. (Correctif du bogue #34284186)<br>• Dans les instructions préparées, certains types de sous-requête pouvaient causer un arrêt du serveur. (Bogue n° 33100586) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL : 10-25 (version 2.11.0, compatible avec MySQL 5.7.12) - Version RDS Extended Support | 2.11.0 | • Correction d'un problème causé par le code de lecture des informations sur le jeu de caractères issues des tables d'événements d'une instruction du schéma de performance (par exemple, events_statements_current) qui n'empêche pas l'écriture simultanée de ces informations sur le jeu de caractères. Le jeu de caractères du texte des requêtes SQL pouvait alors être invalide, ce qui pouvait entraîner un arrêt du serveur. Avec ce correctif, un jeu de caractères non valide entraîne la troncature de la colonne SQL_TEXT et empêche l'arrêt du serveur. (Bogue n° 23540008)<br>• InnoDB : rétroportage d'un correctif pour les bogues n ° 25189192 et 84038. Correction d'un problème de mise à jour de la table du dictionnaire de données INNODB_SY S_DATAFILES par InnoDB après une opération RENAME TABLE qui déplaçait une table vers un autre schéma. Une erreur au redémarrage indiquait alors que le fichier de données du tablespace était introuvable.<br>• InnoDB: correction d'un problème lors duquel le serveur supprimait un index de clé étrangère défini en interne lors de l'ajout d'un nouvel index, et tentait d'utiliser un index secondaire défini sur une colonne virtuelle générée en tant qu'index de clé étrangère, provoquant ainsi l'arrêt du serveur. InnoDB permet désormais à une contrainte de clé étrangère de référencer un index secondaire défini sur une colonne générée virtuelle. (Bogue n° 23533396)<br>• Correction d'un problème d'interblocage engendré par l'exécution simultanée par deux sessions d'une opération INSERT... ON DUPLICATE KEY UPDATE. Lors de l'annulat ion partielle d'un tuple, une autre session pouvait le mettre à jour. La correction de ce bogue annule les correctifs des bogues n° 11758237, 17604730 et 20040791. (Bogue n ° 25966845) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| | | • Correction d'un problème d'octroi inapproprié des privilèges EXECUTE et ALTER ROUTINE aux créateurs de routines, même lorsque automatic_sp_privileges est activé. (Bogue n ° 27407480)<br>• Rétroportage d'un correctif pour le bogue n°24671968 : correction d'un problème lors duquel une requête pouvait produire des résultats incorrects si la clause WHERE contenait une sous-requête dépendante, si la table comportai t un index secondaire sur les colonnes de la liste de sélection , suivi des colonnes de la sous-requête, et si `GROUP BY` ou `DISTINCT` autorisait la requête à utiliser un scan restreint d'index.<br>• Correction d'un problème d'interruption de la réplication si une instruction de suppression de plusieurs tables est émise sur plusieurs tables comportant des clés étrangères. (Bogue n ° 80821)<br>• Correction d'un problème lors duquel, dans des cas particuli ers, certaines erreurs d'esclaves n'étaient pas ignorées même lorsque slave_skip_errors était activé. En cas d'échec de l'ouverture et du verrouillage d'une table ou en cas d'échec des conversions de champs sur un serveur exécutant une réplication basée sur des lignes, l'erreur est considérée comme critique et l'état de slave_skip_errors est ignoré. Le correctif garantit que, lorsque slave_skip_errors est activé, toutes les erreurs signalées lors de l'application d'une transaction sont correctement traitées. (Bogue n° 70640 et 17653275)<br>• Correction d'un problème de réplication d'une instruction `SET PASSWORD` à partir d'un maître MySQL 5.6 vers un esclave MySQL 5.7, ou à partir d'un maître MySQL 5.7 avec la variable système log_builtin_as_identified_by_password définie sur ON vers un esclave MySQL 5.7. Le hachage du |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | mot de passe était lui-même haché avant d'être stocké sur l'esclave. Le problème est maintenant résolu et le hachage du mot de passe répliqué est stocké tel qu'il a été initialement transmis à l'esclave. (Bogue n° 24687073)<br>• Correction d'un problème lors duquel la sérialisation d'une valeur JSON consistant en un sous-document volumineux enveloppé dans plusieurs niveaux de tableaux JSON et/ou d'objets nécessitait parfois un temps d'exécution excessif. (Bogue n° 23031146)<br>• Les instructions qui ne peuvent pas être analysées (en raison, par exemple, d'erreurs de syntaxe) ne sont plus écrites dans le journal des requêtes lentes. (Bogue n° 33732907) |
| Mises à jour du moteur de base de données Aurora MySQL du 01/11/2021 (version 2.10.3) (obsolète) | 2.10.3 | • Correction d'un problème causé par le code de lecture des informations sur le jeu de caractères issues des tables d'événements d'une instruction du schéma de performance (par exemple, events_statements_current) qui n'empêche pas l'écriture simultanée de ces informations sur le jeu de caractères. Le jeu de caractères du texte des requêtes SQL pouvait alors être invalide, ce qui pouvait entraîner un arrêt du serveur. Avec ce correctif, un jeu de caractères non valide entraîne la troncature de la colonne SQL_TEXT et empêche l'arrêt du serveur. (Bogue n° 23540008)<br>• Correction d'un problème d'arrêt du serveur lorsqu'une opération UPDATE nécessitait une table temporaire doté d'une clé primaire de plus de 1 024 octets et que cette table était créée à l'aide d'InnoDB. (Bogue n° 25153670) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL du 21/01/2022 (version 2.10.2) (obsolète) | 2.10.2 | • Correction d'un problème dans InnoDB où une erreur de code liée aux statistiques de table déclenchait une assertion dans le fichier source dict0stats.cc (http://dict0stats.cc/). (Bogue n °24585978)<br>• Un index secondaire sur une colonne virtuelle a été corrompu lorsque l'index a été créé en ligne. Pour les instructions UPDATE (https://dev.mysql.com/doc/refman/5.7/en/update.html), nous corrigeons ce problème comme suit : si la valeur de la colonne virtuelle de l'enregistrement d'index est définie sur NULL, nous générons cette valeur à partir de l'enregistrement d'index du cluster. (Bogue n°30556595)<br>• ASSERTION « !OTHER_LOCK » IN LOCK_REC_ ADD_TO_QUEUE (bogue n°29195848)<br>• HANDLE_FATAL_SIGNAL (SIG=11) DANS __STRCHR_ (Bug #28653104) SSE2<br>• Correction d'un problème pour lequel une interruption de requête pendant un temps d'attente de verrouillage pouvait entraîner une erreur dans InnoDB. (bogue n°28068293)<br>• Les transactions entrelacées pouvaient parfois bloquer l'applicateur de réplica lorsque le niveau d'isolement des transactions était défini sur REPEATABLE READ. (Bogue n ° 25040331)<br>• Correction d'un problème qui pouvait entraîner le blocage des réplicas de journaux binaires en raison du délai d'attente du verrouillage. (bogue n°27189701) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL du 21/10/2021 (version 2.10.1) (obsolète) | 2.10.1 | CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER. (Bogue n° 25209512) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 25/05/2021 (version 2.10.0) (obsolète) | 2.10.0 | • Les transactions entrelacées pouvaient parfois bloquer l'applicateur de réplica lorsque le niveau d'isolement des transactions était défini sur REPEATABLE READ. (Bogue n° 25040331)<br>• Lorsqu'une procédure stockée contenait une instruction faisant référence à une vue qui, à son tour, faisait référence à une autre vue, la procédure n'a pas pu être invoquée avec succès plus d'une fois. (Bogue n° 87858 et n° 26864199)<br>• Pour les requêtes comportant de nombreuses conditions `OR`, l'optimiseur est désormais plus efficace en mémoire et moins susceptible de dépasser la limite de mémoire imposée par la variable système range_optimizer_max_mem_size. En outre, la valeur par défaut de cette variable est passée de 1 536 000 à 8 388 608. (Bogue n° 79450 et n° 22283790)<br>• Réplication : dans la fonction `next_event()` , appelée par le thread SQL d'un réplica pour lire l'événement suivant à partir du journal de relais, le thread SQL ne libérait pas le `relaylog.log_lock` acquis lorsqu'il rencontrait une erreur (par exemple, en raison d'un journal de relais fermé). Par conséquent, tous les autres threads devaient attendre d'obtenir un verrou sur le journal du relais à raccrocher. Avec ce correctif, le verrou est libéré avant que le thread SQL ne quitte la fonction concernée. (Bogue n° 21697821)<br>• Correction d'une corruption de mémoire pour `ALTER TABLE` avec une colonne virtuelle. (Bogue n° 24961167 et n° 24960450)<br>• Réplication : les réplicas multithreads ne pouvaient pas être configurés avec de petites files d'attente à l'aide de slave_pending_jobs_size_max s'ils avaient besoin de traiter des transactions supérieures à cette taille. Tout paquet plus grand que slave_pending_jobs_size_max était rejeté avec |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | l'erreur ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX , même si le paquet était inférieur à la limite fixée par slave_max_allowed_packet. Avec ce correctif, slave_pending_jobs_size_max devient une limite flexible plutôt qu'une limite stricte. Si la taille d'un paquet dépasse slave_pending_jobs_size_max, mais est inférieure à slave_max_allowed_packet, la transaction est conservée jusqu'à ce que tous les workers de réplica aient des files d'attente vides, puis traitées. Toutes les transactions suivantes sont conservées jusqu'à ce que la transaction volumineuse soit terminée. La taille de la file d'attente des workers de réplica peut donc être limitée tout en autorisant des transactions occasionnelles plus importantes. (Bogue n° 21280753 et n° 77406)<br>• Réplication : lors de l'utilisation d'un réplica multithread, les erreurs d'application affichaient des données d'ID de worker incohérentes avec les données externalisées dans les tables de réplique Schéma de performances. (Bogue n° 25231367)<br>• Réplication : Sur une réplique de réplication basée sur GTID exécutée avec -GTID-Mode=ON, -LOG-bin=OFF et utilisant -, lorsqu'une erreur rencontrée qui devait être ignorée n'était pas correctement mise à slave-skip-errors jour, ce qui entraînait une perte de synchronisation avec. Exec_Master_Log_Pos Exec_Master_Log_Pos Read_master_log_pos Si un GTID_NEXT n'était pas spécifié, le réplica ne mettrait jamais à jour son état de GTID lors du retour à partir d'une transaction d'instruction unique. Le Exec_Master_Log_Pos ne serait pas mis à jour, car même si la transaction était terminée, son état GTID indiquerait le contraire. Le correctif supprime la restriction de la mise à jour de l'état GTID lorsqu'une transaction est annulée uniquement si GTID_NEXT est spécifié. (Bogue n° 22268777) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | • Réplication : une instruction partiellement échouée ne consommait pas correctement un GTID généré automatiq uement ou spécifié lorsque la journalisation binaire était désactivée. Ce correctif garantit qu'un DROP TABLE, un DROP USER ou un DROP VIEW ayant partiellement échoué consomment respectivement le GTID correspondant et l'enregistrent dans les tables `@@GLOBAL.GTID_EXECUTED` et `mysql.gtid_executed` lorsque la journalisation binaire est désactivée. (Bogue n° 21686749)<br>• Réplication : les réplicas exécutant MySQL 5.7 ne pouvaient pas se connecter à une source MySQL 5.5 en raison d'une erreur lors de la récupération de server_uuid, qui ne faisait pas partie de MySQL 5.5. Cela était causé par des changements dans la méthode de récupération du `server_uuid` . (Bogue n° 22748612)<br>• Réplication du journal binaire : le mécanisme d'omission de transaction GTID ne fonctionnait pas correctement pour la transaction XA avant ce correctif. Le serveur dispose d'un mécanisme permettant d'ignorer (silencieusement) une transaction GTID s'il a déjà exécuté cette transaction particuli ère par le passé. (Bogue n° 25041920)<br>• Les instructions XA ROLLBACK qui avaient échoué en raison d'un ID de transaction incorrect pouvaient être enregistrées dans le journal binaire avec le bon ID de transaction et donc être traitées par des réplicas de réplication. Une vérification est maintenant effectuée pour la situation d'erreur avant que la journalisation binaire n'ait lieu et les instructions `ROLLBACK` XA échouées ne sont pas consignées. (Bogue n° 26618925)<br>• Réplication : si une réplique a été configurée à l'aide d'une instruction CHANGE MASTER TO qui ne spécifiait ni le nom du fichier journal source ni la position du journal source, puis qu'elle était arrêtée avant que START SLAVE ne soit émise, |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| | | puis redémarrée avec l'option - relay-log-recovery set, la réplication n'a pas démarré. Cela se produisait parce que le thread du récepteur n'avait pas été démarré avant la tentative de récupération du journal de relais. Par conséquent, aucun événement de rotation du journal n'était disponible dans le journal de relais pour fournir le nom du fichier journal source et la position du journal source. Dans ce cas, le réplica ignore désormais la récupération du journal de relais et consigne un avertissement, puis démarre la réplication. (Bogue n ° 28996606 et n° 93397)<br>• Réplication : dans la réplication basée sur les lignes, un message qui affichait mal les longueurs de champ était renvoyé lors de la réplication à partir d'une table comportan t une colonne `utf8mb3` vers une table de la même définition que celle où la colonne était définie avec un jeu de caractères `utf8mb4`. (Bogue n° 25135304 et n° 83918)<br>• Réplication : lorsqu'une instruction RESET SLAVE était émise sur une réplique de réplication GTIDs en cours d'utilisation, les fichiers journaux de relais existants étaient purgés, mais le nouveau fichier journal de relais de remplacement était généré avant que l'ensemble des fichiers reçus GTIDs pour le canal n'ait été effacé. L'ancien ensemble GTID a donc été écrit dans le nouveau fichier journal du relais en tant qu'`PREVIOUS_ GTIDS` événement, provoquant une erreur fatale lors de la réplication indiquant que la réplique contenait GTIDs plus que la source, même si l'ensemble gtid_executed pour les deux serveurs était vide. Désormais, lorsqu'il `RESET SLAVE` est émis, l'ensemble des données reçues GTIDs est effacé avant que le nouveau fichier journal de relais ne soit généré, de sorte que cette situation ne se produise pas. (Bogue n ° 27411175) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| | | • Réplication : GTIDs lors de l'utilisation pour la réplication, les transactions comprenant des instructions à l'origine d'une erreur d'analyse (ER_PARSE_ERROR) ne pouvaient pas être ignorées manuellement par la méthode recommandée consistant à injecter une transaction vide ou de remplacement avec le même GTID. Cette action doit permettre au réplica d'identifier le GTID comme déjà utilisé et donc d'ignorer la transaction indésirable qui a partagé son GTID. Toutefois, dans le cas d'une erreur d'analyse, parce que l'instruction a été analysée avant que le GTID ne soit vérifié pour voir s'il devait être ignoré, le thread qui a appliqué la réplication s'est arrêté en raison de l'erreur d'analyse, même si l'intention était d'ignorer la transaction malgré tout. Avec ce correctif, le thread qui a appliqué la réplication ignore désormais les erreurs d'analyse si la transaction concernée doit être ignorée, car le GTID était déjà utilisé. Notez que ce changement de comportement ne s'applique pas dans le cas d'applications constituées d'une sortie de journal binaire produite par `mysqlbinlog` . Dans ce cas, il y aurait un risque qu'une transaction comportant une erreur d'analyse qui suit immédiatement une transaction ignorée soit également ignorée, alors qu'elle devrait générer une erreur. (Bogue n° 27638268)<br>• Réplication : activez le thread SQL pour que GTID ignore une transaction partielle. (Bogue n° 25800025)<br>• Réplication : lorsqu'un paramètre de délai d'expiration négatif ou fractionnaire avait été fourni à `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()` , le serveur se comportait de manière inattendue. Avec ce correctif :<br>  • une valeur de délai d'expiration fractionnaire est lue telle quelle, sans arrondi ; |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | • une valeur de délai d'expiration négatif est rejetée avec une erreur si le serveur est en mode SQL strict. Si le serveur n'est pas en mode SQL strict, la valeur fait en sorte que la fonction renvoie `NULL` immédiatement et sans attendre, puis émette un avertissement. (Bogue n° 24976304 et n° 83537)<br>• Réplication : si la fonction `WAIT_FOR_EXECUTED_GTID_SET()` avait été utilisée avec une valeur de délai d'expiration incluant une partie fractionnée (par exemple, 1,5), une erreur dans la logique de conversion de types signifiait que le délai d'expiration était arrondi à la seconde entière la plus proche et à zéro pour les valeurs inférieures à 1 seconde (par exemple, 0,1). La logique de conversion de types a maintenant été corrigée de sorte que la valeur du délai d'expiration soit appliquée comme indiqué initialement sans arrondi. Merci à Dirkjan Bussink pour sa contribution. (Bogue n° 29324564 et n° 94247)<br>• GTIDs Lorsque cette option est activée, XA COMMIT sur une transaction XA déconnectée dans le cadre d'une transacti on à instructions multiples a généré une assertion. (Bogue n ° 22173903)<br>• Réplication : une assertion était déclenchée dans les builds de débogage si une instruction XA ROLLBACK était émise pour un identifiant de transaction inconnu lorsque la valeur gtid_next avait été définie manuellement. Le serveur ne tente plus de mettre à jour l'état du GTID si une instruction `ROLLBACK` XA échoue avec une erreur. (Bogue n° 27928837 et n° 90640)<br>• Correction d'un problème d'ordre de tri erroné lorsque plusieurs fonctions `CASE` sont utilisées dans la clause `ORDER BY`. (Bogue n° 22810883) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | • Certaines requêtes utilisant le tri pouvaient accéder à une colonne non initialisée lors de l'optimisation et provoquer la fermeture du serveur. (Bogue n° 27389294) |
| Mises à jour du moteur de base de données Aurora MySQL du 12/11/2022 (version 2.09.3) (obsolète) | 2,09.3 | • ASSERTION !M_PREBUILT->TRX->CHECK_FOREIGNS. (Bogue n° 23533396)<br>• Réplication :* dans certaines circonstances, un problème de verrouillage dans la fonction WAIT_FOR_EXECUTED_GTID_SET () peut entraîner le blocage du serveur. Ce problème a été corrigé. (Bogue n° 29550513) |
| Mises à jour du moteur de base de données Aurora MySQL du 11/12/2020 (version 2.09.1) (obsolète) | 2.09.1 | • Réplication : les transactions entrelacées pouvaient parfois bloquer l'applicateur esclave lorsque le niveau d'isolement des transactions était défini sur REPEATABLE READ. (Bogue n° 25040331)<br>• Pour les tables ayant une colonne TIMESTAMP ou DATETIME avec une valeur CURRENT_TIMESTAMP par défaut, la colonne pouvait être initialisée à `0000-00-00 00:00:00` si la table avait un déclencheur `BEFORE INSERT`. (Bogue n° 25209512, Bogue n° 84077)<br>• Pour les instructions INSERT pour lesquelles la liste `VALUES` produisait des valeurs pour la deuxième ligne ou une ligne suivante à l'aide d'une sous-requête contenant une jointure, le serveur pouvait s'arrêter s'il n'avait pas réussi à résoudre les privilèges requis. (Bogue n° 23762382) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL du 12/11/2022 (version 2.08.3) (obsolète) | 2.08.3 | • Bogue n° 23762382 - INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR.<br>• Bogue n° 25209512 - CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER. |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 06/02/2020 (version 2.08.0) (obsolète) | 2.08.0 | • Bogue #25289359 : un verrou de cache de texte intégral pris lors de la synchronisation des données n'était pas libéré si la taille du cache de texte intégral dépassait la limite de taille du cache de texte intégral.<br>• Bogue #29138644 : la modification manuelle de l'heure système pendant l'exécution du serveur MySQL a provoqué des retards du thread de nettoyage de page.<br>• Bogue #25222337 : un nom de champ de colonne virtuelle NULL dans un index virtuel a provoqué une sortie du serveur lors d'une comparaison de noms de champ qui se produit lors du remplissage de colonnes virtuelles affectées par une contrainte de clé étrangère.<br>• Bogue #25053286 : l'exécution d'une procédure stockée contenant une requête qui a accédé à une vue pouvait allouer de la mémoire qui n'était pas libérée jusqu'à la fin de la session.<br>• Bogue #25586773 : l'exécution d'une procédure stockée contenant une instruction qui a créé une table à partir du contenu de certaines instructions SELECT pouvait entraîner une fuite de mémoire.<br>• Bogue #28834208 : au cours de l'application de journal, après une opération OPTIMISTIE TABLE, InnoDB n'a pas rempli de colonnes virtuelles avant de vérifier les mises à jour d'index de colonne virtuelle.<br>• Bogue #26666274 : boucle infinie dans le conteneur de mémoire tampon de schéma de performance en raison d'un dépassement d'entier non signé 32 bits. |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 16/06/2022 (version 2.07.8) (obsolète) | 2,07.8 | Lorsqu'une mise à jour nécessitait une table temporaire avec une clé primaire de plus de 1 024 octets et que cette table était créée à l'aide d'InnoDB, le serveur pouvait s'arrêter. (Bogue n ° 25153670) |
| Mises à jour du moteur de base de données Aurora MySQL du 02/09/2021 (version 2.07.6) (obsolète) | 2,07.6 | • L'INSERTION D'ENREGISTREMENTS DE TAILLE 64K PREND TROP DE TEMPS. (Bogue n° 23031146) |
| Mises à jour du moteur de base de données Aurora MySQL du 04/03/2021 (version 2.07.) (obsolète) | 2.07.4 | • Correction d'un problème dans l'analyseur n-gram de texte intégral lors de la gestion de jetons contenant « » (espace), « % » ou « , ». Les clients doivent reconstruire leurs index FTS si vous utilisez un analyseur n-gram. (Bogue n ° 25873310)<br>• Correction d'un problème qui pouvait entraîner le redémarra ge du moteur pendant l'exécution de la requête avec des vues SQL imbriquées. (Bogue n° 27214153, Bogue n° 26864199) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL du 10/11/2020 (version 2.07.3) (obsolète) | 2.07.3 | • InnoDB : les transactions XA simultanées exécutées avec succès pour la phase de préparation XA sur le maître étaient en conflit lorsqu'elles étaient réutilisées sur l'esclave , entraînant ainsi un délai d'attente de verrouillage dans le thread applicateur. Le conflit était dû à la plage de verrouill age d'écart qui variait lorsque les transactions étaient réutilisé es en série sur l'esclave. Pour éviter ce type de conflit, les verrous d'écart acceptés par les transactions XA au niveau d'isolement READ COMMITTED sont désormais publiés (et ne sont plus hérités) lorsque les transactions XA atteignen t l'étape de préparation. (Bogue n° 27189701, Bogue n ° 25866046)<br>• InnoDB : un verrou d'écart a été pris inutilement lors de la validation de clé étrangère et de l'utilisation du niveau d'isolement READ COMMITTED. (Bogue n° 25082593)<br>• Réplication : lors de l'utilisation de transactions XA, si un délai d'attente de verrouillage ou un blocage se produisait pour le thread applicateur (SQL) sur un esclave de réplication, la nouvelle tentative automatique ne fonctionnait pas, car même si le thread SQL procédait à une restauration, il ne restaurai t pas la transaction XA. Par conséquent, lorsque la transacti on était relancée, le premier événement était XA START et n'était pas valide étant donné que la transaction XA était déjà en cours, entraînant une erreur XAER_RMFAIL. (Bogue n ° 24764800)<br>• Réplication : les transactions entrelacées pouvaient parfois bloquer l'applicateur esclave lorsque le niveau d'isolement des transactions était défini sur REPEATABLE READ. (Bogue n ° 25040331)<br>• Réplication : la valeur renvoyée par une instruction SHOW SLAVE STATUS pour la taille combinée totale de tous les fichiers journaux relais existants (Relay_Log_Space) pouvait |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | devenir beaucoup plus élevée que l'espace disque réel utilisé par les journaux relais. Le I/O thread ne verrouillait pas la variable lors de la mise à jour de la valeur, de sorte que le thread SQL pouvait automatiquement supprimer un fichier journal de relais et écrire une valeur réduite avant que le I/O thread n'ait fini de mettre à jour la valeur. Le thread d'I/O écrivait ensuite son calcul de taille d'origine, en ignorant la mise à jour du thread SQL et en rajoutant ainsi l'espace du fichier supprimé. La valeur Relay_Log_Space est désormais verrouillée pendant les mises à jour afin d'empêcher les mises à jour simultanées et de garantir un calcul précis. (Bogue n° 26997096, Bogue n° 87832)<br>• Pour les instructions INSERT pour lesquelles la liste VALUES produisait des valeurs pour la deuxième ligne ou une ligne suivante à l'aide d'une sous-requête contenant une jointure, le serveur pouvait s'arrêter s'il n'avait pas réussi à résoudre les privilèges requis. (Bogue n° 23762382)<br>• Pour les tables ayant une colonne TIMESTAMP ou DATETIME avec une valeur CURRENT_TIMESTAMP par défaut, la colonne pouvait être initialisée à `0000-00-00 00:00:00` si la table avait un déclencheur `BEFORE INSERT`. (Bogue n° 25209512, Bogue n° 84077)<br>• Les tentatives simultanées de plusieurs threads pour enregistrer et annuler l'enregistrement des objets de schéma de performances de métadonnées pouvaient entraîner l'arrêt d'un serveur. (Bogue n° 26502135)<br>• L'exécution d'une procédure stockée contenant une instruction qui avait créé une table à partir du contenu de certaines instructions SELECT pouvait entraîner une fuite de mémoire. (Bogue n° 25586773)<br>• L'exécution d'une procédure stockée contenant une requête qui avait accédé à une vue pouvait allouer de la mémoire |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | qui n'était pas libérée avant la fin de la séance. (Bogue n ° 25053286)<br>• Certains cas de matérialisation de sous-requête pouvaient provoquer l'arrêt d'un serveur. Ces requêtes produisent maintenant une erreur suggérant que la matérialisation doit être désactivée. (Bogue #26402045)<br>• Les requêtes avec de nombreuses jointures gauches étaient lentes si la mise en mémoire tampon de jointure était utilisée (par exemple, à l'aide de l'algorithme de boucle imbriquée par bloc). (Bogue n° 18898433, Bogue n° 72854)<br>• L'optimiseur ignorait la deuxième colonne d'un index composite lors de l'exécution d'une jointure interne avec une clause `LIKE` sur la deuxième colonne. (Bogue n° 28086754) |
| Mises à jour du moteur de base de données Aurora MySQL du 17/04/2020 (version 2.07.2) (obsolète) | 2.07.2 | • Bogue #23104498 : correction d'un problème dans le schéma de performance lors du signalement de la mémoire totale utilisée. (https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b)<br>• Bogue #22551677 : correction d'un problème dans le schéma de performance qui pouvait entraîner le plantage du moteur de base de données lorsque d'une tentative de le mettre hors ligne. (https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9)<br>• Bogue #23550835, Bogue #23298025, Bogue #81464 : correction d'un problème dans le schéma de performance qui provoquait un plantage du moteur de base de données en raison du dépassement de la capacité d'un tampon interne. (https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc, https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 2.07.0) (obsolète) | 2.07.0 | • Bogue #26251621 : COMPORTEMENT INCORRECT DU DÉCLENCHEUR ET DE GCOL<br>• Bug #22574695: ASSERTION `!TABLE \|\| (!TABLE-> READ_SET \|\| BITMAP_IS_SET(TABLE->READ_SET, FIEL<br>• Bogue #25966845 : LES INSERTIONS SUR CLÉ DUPLIQUÉE GÉNÈRENT UN VERROU<br>• Bogue #23070734 : LES TABLES TRONQUÉES SIMULTANÉ ES PROVOQUENT UN ARRÊT<br>• Bogue #26191879 : LES CASCADES DE CLÉS ÉTRANGÈRES UTILISENT TROP DE MÉMOIRE<br>• Bogue #20989615: INNODB AUTO_INCREMENT GÉNÈRE DEUX FOIS LA MÊME VALEUR |
| Mises à jour du moteur de base de données Aurora MySQL du 11/11/2019 (version 2.05.0) (obsolète) | 2.05.0 | • Bug #23054591 : PURGE BINARY LOGS TO lit l'intégralité du fichier binlog et provoque MySql un blocage |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 14/08/2020 (version 2.04.9) (obsolète) | 2.04.9 | • Bug #23070734, Bug #80060 : la fonction TRUNCATE simultanée TABLEs provoque des blocages<br>• Bogue n° 23103937 : PS_TRUNCATE_ALL_TABLES() NE FONCTIONNE PAS EN MODE SUPER_READ_ONLY<br>• Bogue n° #22551677 : lorsque vous mettez le serveur hors connexion, une condition de concurrence dans le schéma de performances peut entraîner la sortie du serveur.<br>• Bogue n° 27082268 : synchronisation FTS non valide.<br>• BOGUE n° 12589870 : correction d'un problème qui provoquait un redémarrage avec une instruction multi-requête lorsque le cache de requête est activé.<br>• Bogue n° 26402045 : certains cas de matérialisation de sous-requête peuvent provoquer la sortie d'un serveur. Ces requêtes produisent maintenant une erreur suggérant que la matérialisation doit être désactivée.<br>• Bogue n° 18898433 : les requêtes avec de nombreuses jointures gauche étaient lentes si la mise en mémoire tampon de jointure était utilisée (par exemple, en utilisant l'algorithme de boucle imbriquée par bloc).<br>• Bogue n° 25222337 : un nom de champ de colonne virtuelle NULL dans un index virtuel a provoqué une sortie du serveur lors d'une comparaison de noms de champ qui se produit lors du remplissage de colonnes virtuelles affectées par une contrainte de clé étrangère. (https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131)<br>• Bogue n° 25053286 : l'exécution d'une procédure stockée contenant une requête qui a accédé à une vue pouvait allouer de la mémoire qui n'était pas libérée jusqu'à la fin de la session. (https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | • Bug #25586773 : L'exécution d'une procédure stockée contenant une instruction qui a créé une table à partir du contenu de certaines instructions SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) peut entraîner une fuite de mémoire. (https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1)<br>• Bogue n° 26666274 : BOUCLE INFINITE DANS LE CONTAINER DE MÉMOIRE TAMPON DE SCHÉMA DE PERFORMANCE.<br>• Bogue n° 23550835, Bogue n° 23298025, Bogue n° 81464 : une table SELECT Performance Schema lorsqu'un tampon interne était plein pouvait provoquer la sortie d'un serveur. |
| Mises à jour du moteur de base de données Aurora MySQL du 19/09/2019 (version 2.04.6) (obsolète) | 2.04.6 | • Bug #23054591 : PURGE BINARY LOGS TO lit l'intégralité du fichier binlog et provoque MySql un blocage |
| Mises à jour du moteur de base de données Aurora MySQL du 02/05/2019 (version 2.04.2) (obsolète) | 2.04.2 | Bogue n°24829050 - INDEX_MERGE_INTERSECTION OPTIMIZATION CAUSES WRONG QUERY RESULTS |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL du 11/10/2018 (version 2.03) (obsolète) | 2.03 | • UNE ANALYSE EN ORDRE INVERSE SUR UNE TABLE PARTITIONNÉE DONNE UN CLASSEMENT ICP PAR DESCRIPTION (bogue n° 24929748).<br>• JSON_OBJECT CRÉE UN CODE JSON NON VALIDE (bogue n° 26867509).<br>• L'INSERTION D'UN GROS VOLUME DE DONNÉES JSON PREND UNE DURÉE INHABITUELLE (bogue n° 22843444).<br>• LES TABLES PARTITIONNÉES UTILISENT DAVANTAGE DE MÉMOIRE DANS LA VERSION 5.7 QUE DANS LA VERSION 5.6 (bogue n° 25080442). |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 21/09/2018 (version 2.02.4) (obsolète) | 2.02.4 | • BUG#13651665 INNODB MAY BE UNABLE TO LOAD TABLE DEFINITION AFTER RENAME<br>• BUG#21371070 INNODB: CANNOT ALLOCATE 0 BYTES.<br>• BUG#21378944 FTS ASSERT ENC.SRC_ILIST_PTR != NULL, FTS_OPTIMIZE_WORD(), OPTIMIZE TABLE<br>• BUG#21508537 ASSERTION FAILURE UT_A(!VIC TIM_TRX->READ_ONLY)<br>• BUG#21983865 UNEXPECTED DEADLOCK WITH INNODB_AUTOINC_LOCK_MODE=0<br>• BUG#22679185 INVALID INNODB FTS DOC ID DURING INSERT<br>• BUG#22899305 GCOLS: ASSERTION: !(COL->PR TYPE & 256).<br>• BUG#22956469 MEMORY LEAK INTRODUCED IN 5.7.8 IN MEMORY/INNODB/OS0FILE<br>• BUG#22996488 CRASH IN FTS_SYNC_INDEX WHEN DOING DDL IN A LOOP<br>• BUG#23014521 GCOL:INNODB: ASSERTION: !IS_V<br>• BUG#23021168 REPLICATION STOPS AFTER TRX IS ROLLED BACK ASYNC<br>• BUG#23052231 ASSERTION: ADD_AUTOINC < DICT_TABLE_GET_N_USER_COLS<br>• BUG#23149683 ROTATE INNODB MASTER KEY WITH KEYRING_OKV_CONF_DIR MISSING: SIGSEGV; SIGNAL 11<br>• BUG#23762382 INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR<br>• BUG#25209512 CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | • BUG#26626277 BUG IN "INSERT... ON DUPLICATE KEY UPDATE" QUERY<br>• BUG#26734162 INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE<br>• BUG#27460607 INCORRECT WHEN INSERT SELECT's SOURCE TABLE IS EMPTY |
| Mises à jour du moteur de base de données Aurora MySQL du 03/05/2018 (version 2.02) (obsolète) | 2.02.0 | La jointure gauche retourne des résultats incorrects du côté extérieur (Bogue n° 22833364) |

## Bogues MySQL corrigés par les mises à jour du moteur de base de données Aurora MySQL 1.x

La version d'Aurora compatible avec MySQL 5.6 contient toutes les corrections de bogues effectuées par MySQL 5.6.10. Le tableau suivant identifie les bogues MySQL supplémentaires qui ont été corrigés par les mises à jour du moteur de base de données Aurora MySQL, ainsi que la mise à jour dans laquelle ils ont été corrigés.

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 18/03/2021 | 1.23.2 | • Réplication : pendant l'exécution d'une instruction SHOW BINLOG EVENTS, toute transaction parallèle a été bloquée. Le correctif garantit que le processus SHOW BINLOG EVENTS n'acquiert désormais un verrou que pendant la durée du calcul de la position finale du fichier ; par conséquent, les transacti |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| (version 1.23.2) (obsolète) | | ons parallèles ne sont pas bloquées pendant de longues durées. (Bogue n° 76618, Bogue n° 20928790) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 02/09/2020 (version 1.23.0) (obsolète) | 1.23.0 | • Les événements Binlog avec `ALTER TABLE ADD COLUMN ALGORITHM=QUICK` seront réécrits en tant que `ALGORITHM =DEFAULT` de manière à être compatibles avec l'édition de la communauté.<br>• BOGUE #22350047 : SI LE CLIENT EST TUÉ APRÈS LA RESTAURATION AU POINT DE SAUVEGARDE PRÉCÉDENT STMTS VALIDÉ<br>• Bogue #29915479 : L'EXÉCUTION DE COM_REGIS TER_SLAVE SANS COM_BINLOG_DUMP PEUT GÉNÉRER L'ARRÊT DU SERVEUR<br>• Bogue #30441969 : BOGUE #29723340 : INCIDENT MYSQL SERVER APRÈS REQUÊTE SQL AVEC ?AST DE DONNÉES<br>• Bogue #30628268 : INCIDENT DE MÉMOIRE INSUFFISA NTE<br>• Bogue #27081349 : COMPORTEMENT INATTENDU LORS D'UNE SUPPRESSION AVEC FONCTION SPATIALE<br>• Bogue #27230859 : COMPORTEMENT INATTENDU LORS DE LA GESTION D'UN POLYGONE NON VALIDE<br>• Bogue #27081349 : COMPORTEMENT INATTENDU LORS DE LA SUPPRESSION AVEC FONCTION SPATIALE<br>• Bogue #26935001 : AUTO_INCREMENT DE MODIFICAT ION DE TABLE TENTE DE LIRE L'INDEX À PARTIR D'UN ESPACE DE TABLE SUPPRIMÉ<br>• Bogue #29770705 : INCIDENT SERVEUR PENDANT L'EXÉCUTION DE SELECT AVEC CLAUSE WHERE SPÉCIFIQUE<br>• Bogue #27659490 : SELECT AVEC UTILISATION DE PLAGE DYNAMIQUE ET DE FUSION D'INDEX NÉCESSITE TROP DE MÉMOIRE (MÉMOIRE INSUFFISANTE) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | • Bogue #24786290 : INTERRUPTION DE LA RÉPLICATI ON LORSQUE LE BOGUE #74145 SE PRODUIT SUR LE PRINCIPAL<br>• Bogue #27703912 : UTILISATION DE MÉMOIRE EXCESSIVE AVEC NOMBREUSES PRÉPARATIONS<br>• Bug #20527363 : CRASH DE LA TABLE TEMPORAIRE TRUNCATE : ! TF2DICT_ _FLAG_IS_SET (TABLE, DICT_ _TEMPORAIRE) TF2<br>• Bogue #23103937 : PS_TRUNCATE_ALL_TABLES() NE FONCTIONNE PAS EN MODE SUPER_READ_ONLY<br>• Bogue #25053286 : L'UTILISATION DE VUE AVEC CONDITION DANS LA PROCÉDURE GÉNÈRE UN COMPORTEMENT INCORRECT (corrigé dans 5.6.36)<br>• Bogue #25586773 : COMPORTEMENT INCORRECT POUR SÉLECTION DE TABLE DE CRÉATION DANS UNE BOUCLE DANS SP (corrigé dans 5.6.39)<br>• Bogue #27407480 : AUTOMATIC_SP_PRIVILEGES NÉCESSITE LES PRIVILÈGES INSERT POUR LA TABLE MYSQL.USER<br>• Bogue #26997096 : la valeur de `relay_log_space` n'est pas mise à jour de manière synchronisée, de sorte qu'elle est parfois beaucoup plus élevée que l'espace disque réel utilisé par les journaux relais.<br>• Bogue #15831300 SLAVE_TYPE_CONVERSIONS=ALL_ NON_LOSSY NE FONCTIONNE PAS COMME PRÉVU<br>• Bogue backport bogue SSL #17087862, bogue #20551271<br>• Bogue #16894092 : RÉGRESSION DE PERFORMANCE DANS 5.6.6+ POUR INSERT INTO ... SELECT ... FROM (corrigé dans 5.6.15). |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | • Porter un correctif de bogue lié à `SLAVE_TYPE_CONVERSIONS` . |
| Mises à jour du moteur de base de données Aurora MySQL du 19/11/2020 (version 1.22.3) (obsolète) | 1.22.3 | • Bogue n° 26654685 : Un ID d'index corrompu rencontré lors d'une vérification de clé étrangère déclenchait une assertion.<br>• Bogue n° 15831300 : Par défaut, lors de la promotion de nombres entiers d'un type plus petit sur le maître à un type plus grand sur l'esclave (par exemple, d'une colonne SMALLINT sur le maître à une colonne BIGINT sur l'esclave ), les valeurs promues sont traitées comme si elles étaient signées. Dans de tels cas, il est possible de modifier ou de remplacer ce comportement à l'aide de `ALL_SIGNED` , de `ALL_UNSIGNED` ou des deux dans l'ensemble des valeurs spécifiées pour la variable système serveur slave_type_conversions. Pour en savoir plus, consultez la section Row-based replication: attribute promotion and demotion, ainsi que la description de la variable.<br>• Bogue n° 17449901 : Avec `foreign_key_checks=0` , InnoDB permettait de supprimer un index requis par une contrainte de clé étrangère, plaçant la table dans une incohérence et provoquant l'échec de la vérification de la clé étrangère lors du chargement de la table. InnoDB empêche désormais de supprimer un index requis par une contrainte de clé étrangère, même avec foreign_key_checks=0. La contrainte de clé étrangère doit être supprimée avant de supprimer l'index de clé étrangère.<br>• BOGUE n° 20768847 : Une opération ALTER TABLE ... Opération DROP INDEX sur une table avec des dépendances de clé étrangère déclenchait une assertion. |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 1.22.0) (obsolète) | 1.22.0 | • Bogue #16346241 - ARRÊT DU SERVEUR DANS ITEM_PARAM::QUERY_VAL_STR<br>• Bogue #17733850 - ARRÊT DE NAME_CONST() DANS ITEM_NAME_CONST::ITEM_NAME_CONST()<br>• Bogue #20989615: INNODB AUTO_INCREMENT GÉNÈRE DEUX FOIS LA MÊME VALEUR<br>• Bogue #20181776 - LE CONTRÔLE D'ACCÈS NE CORRESPOND PAS À L'HÔTE LE PLUS SPÉCIFIQUE QUAND IL CONTIENT DES CARACTÈRES GÉNÉRIQUES<br>• Bogue #27326796 - ARRÊT DE MYSQL AVEC ÉCHEC DE L'ASSERTION INNODB DANS LE FICHIER PARS0PARS.CC<br>• Bug #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL |
| Mises à jour du moteur de base de données Aurora MySQL du 25/11/2019 (version 1.21.0) (obsolète) | 1.21.0 | • Bug #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) DANS __MEMMOVE_ _BACK FROM STRING : :COPY SSSE3<br>• Bogue n° 17059925 : pour les instructions UNION, la valeur examinée par les lignes a été calculée de façon incorrecte. Cela a donné des valeurs trop importantes pour la colonne `ROWS_EXAMINED` des tables de l'instruction Schéma des performances (par exemple, events_statements_current).<br>• Bogue n° 11827369 : certaines requêtes avec des sous-requ êtes imbriquées `SELECT ... FROM DUAL` ont généré une assertion.<br>• Bogue n° 16311231 : des résultats incorrects ont été renvoyés lorsqu'une requête contenait une sous-requête dans une clause `IN` qui contenait une opération XOR dans la clause `WHERE`. |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 11/11/2019 (version 1.20.0) (obsolète) | 1.20.0 | • Bug #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) DANS __MEMMOVE_ _BACK FROM STRING : :COPY SSSE3<br>• Bogue n° 17059925 : pour les instructions UNION, la valeur examinée par les lignes a été calculée de façon incorrecte. Cela a donné des valeurs trop importantes pour la colonne `ROWS_EXAMINED` des tables de l'instruction Schéma des performances (par exemple, events_statements_current).<br>• Bogue n° 11827369 : certaines requêtes avec des sous-requ êtes imbriquées `SELECT ... FROM DUAL` ont généré une assertion.<br>• Bogue n° 16311231 : des résultats incorrects ont été renvoyés lorsqu'une requête contenait une sous-requête dans une clause `IN` qui contenait une opération XOR dans la clause `WHERE`. |
| Mises à jour du moteur de base de données Aurora MySQL du 19/09/2019 (version 1.19.5) (obsolète) | 1.19.5 | • CVE-2018-2696<br>• CVE-2015-4737<br>• Bug #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) DANS __MEMMOVE_ _BACK FROM STRING : :COPY SSSE3<br>• Bogue n° 17059925 : pour les instructions UNION, la valeur examinée par les lignes a été calculée de façon incorrecte. Cela a donné des valeurs trop importantes pour la colonne `ROWS_EXAMINED` des tables de l'instruction Schéma des performances (par exemple, events_statements_current).<br>• Bogue n° 11827369 : certaines requêtes avec des sous-requ êtes imbriquées `SELECT ... FROM DUAL` ont généré une assertion.<br>• Bogue n° 16311231 : des résultats incorrects ont été renvoyés lorsqu'une requête contenait une sous-requête dans une clause `IN` qui contenait une opération XOR dans la clause `WHERE`. |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 07/02/2019 (version 1.19.0) (obsolète) | 1.19.0 | • Bogue n°32917 : DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY<br>• Bogue n°63144 : CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE |
| Mises à jour du moteur de base de données Aurora MySQL du 17/01/2019 (version 1.17.8) (obsolète) | 1.17.8 | • BOGUE 13418638 : CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE |
| Mises à jour du moteur de base de données Aurora MySQL du 08/10/2018 (version 1.17.7) (obsolète) | 1.17.7 | • Une opération Drop Index sur une colonne de clé étrangère entraîne un problème de table manquante. (Bogue n ° 16208542)<br>• Fuite de mémoire dans add_derived_key(). (Bogue n° 76349)<br>• Pour les tables partitionnées, les requêtes peuvent renvoyer différents résultats en fonction de l'utilisation ou non de la fusion d'index. (Bogue n° 16862316)<br>• Les requêtes utilisant l'optimisation index_merge (consultez Index Merge Optimization) peuvent renvoyer des résultats non valides lorsqu'elles sont exécutées sur des tables qui ont été partitionnées par HASH. (Bogue n° 17588348) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 06/09/2018 (version 1.17.6) (obsolète) | 1.17.6 | • Pour une instruction ALTER TABLE qui renommait ou modifiait la valeur par défaut d'une colonne BINARY, l'altérat ion était effectuée en utilisant une copie de la table et non sur place. (Bogues n° 67141, 14735373, 69580 et 17024290)<br>• Une jointure externe entre une table standard et une table dérivée qu'elle regroupe implicitement peut entraîner l'arrêt du serveur. (Bogue n° 16177639) |
| Mises à jour du moteur de base de données Aurora MySQL du 13/03/2018 (version 1.17) (obsolète) | 1.17.0 | • LAST_INSERT_ID n'est pas répliqué correctement si les filtres de réplication sont utilisés (bogue n° 69861)<br>• La requête retourne différents résultats en fonction du paramètre INDEX_MERGE (bogue n° 16862316)<br>• Réexécution de la routine stockée du traitement de requête, plan de requête inefficace (bogue n° 16346367)<br>• InnoDB FTS : assertion dans FTS_CACHE_APPEND_D ELETED_DOC_IDS (bogue n° 18079671)<br>• Assertion RBT_EMPTY(INDEX_CACHE->WORDS) dans ALTER TABLE CHANGE COLUMN (bogue n° 17536995)<br>• La recherche InnoDB en texte intégral ne trouve pas d'enregis trements lorsque des points de sauvegarde sont impliqués (bogue n° 70333, bogue n° 17458835) |
| Mises à jour du moteur de base de données Aurora MySQL du 20/11/2017 (version 1.15.1) (obsolète) | 1.15.1 | • Annulation — Ralentissement de l'instance MySQL « doing SYNC index » (bogue n° 73816)<br>• Annulation — Assertion RBT_EMPTY(INDEX_CACHE- >WORDS) dans ALTER TABLE CHANGE COLUMN (bogue n ° 17536995)<br>• Annulation — La recherche InnoDB en texte intégral ne trouve pas d'enregistrements lorsque des points de sauvegarde sont impliqués (bogue n° 70333) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL du 24/10/2017 (version 1.15) (obsolète) | 1.15.0 | • CREATE USER accepte le plugin et le hachage du mot de passe, mais ignore le hachage du mot de passe (bogue n ° 78033)<br>• Le moteur de partitionnement ajoute des champs à l'ensembl e de bits de lecture pour pouvoir retourner des entrées triées à partir d'un index partitionné. Par conséquent, le tampon de jointure essaie de lire des champs non nécessaires. Corrigé en n'ajoutant pas tous les champs de partitionnement à read_set, mais en triant uniquement les champs à préfixe déjà spécifié de read_set. Ajout d'un DBUG_ASSERT qui implique la lecture du premier champ en cas de key_cmp (bogue n ° 16367691).<br>• Ralentissement de l'instance MySQL « doing SYNC index » (bogue n° 73816)<br>• Assertion RBT_EMPTY(INDEX_CACHE->WORDS) dans ALTER TABLE CHANGE COLUMN (bogue n° 17536995)<br>• La recherche InnoDB Fulltext ne trouve pas d'enregistrements lorsque des points de sauvegarde sont impliqués (bogue n ° 70333) |
| Mises à jour du moteur de base de données Aurora MySQL : 13/03/201 8 (version 1.14.4) (obsolète) | 1.14.4 | • Les événements pouvant être ignorés ne fonctionnent pas et ne sont pas testés (bogue n° 74683)<br>• NEW->OLD ASSERT FAILURE 'GTID_MODE > 0' (Bug #20436436) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL : 07/08/2017 (version 1.14) (obsolète) | 1.14.0 | Une recherche en texte intégral associée aux tables dérivées (sous-requêtes de la clause `FROM`) entraînait un arrêt du serveur. Maintenant, si une opération de texte intégral dépend d'une table dérivée, le serveur produit une erreur indiquant qu'une recherche en texte intégral ne peut pas être effectuée sur une table matérialisée. (Bogues n° 68751 et 16539903) |
| Mises à jour du moteur de base de données Aurora MySQL : 15/05/2017 (version 1.13) (obsolète) | 1.13.0 | • Le rechargement d'une table qui a été expulsée quand elle était vide a provoqué la réinitialisation d'une valeur AUTO_INCREMENT. (Bogues n° 21454472 et 77743)<br>• Un enregistrement d'index a été introuvable à la restauration en raison d'incohérences dans la structure purge_node_t. Celles-ci ont entraîné des messages d'avertissement et d'erreur tels que « erreur dans la mise à jour de l'entrée d'index secondaire », « impossible de purger un enregistrement » et « tentative de purge de l'entrée d'index secondaire non marquée pour la suppression ». (Bogues n° 19138298, 70214, 21126772 et 21065746)<br>• Le calcul incorrect de la taille de la pile de l'opération qsort entraîne un dépassement de la pile. (Bogue n° 73979)<br>• Enregistrement introuvable dans un index à la restauration. (Bogues n° 70214 et 72419)<br>• ALTER TABLE ajoute la colonne TIMESTAMP sur la mise à jour CURRENT_TIMESTAMP insère des données ZERO (Bogue n° 17392) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL : 05/04/2017 (version 1.12) (obsolète) | 1.12.0 | • Le rechargement d'une table qui a été expulsée quand elle était vide a provoqué la réinitialisation d'une valeur AUTO_INCREMENT. (Bogues n° 21454472 et 77743)<br>• Un enregistrement d'index a été introuvable à la restauration en raison d'incohérences dans la structure purge_node_t. Celles-ci ont entraîné des messages d'avertissement et d'erreur tels que « erreur dans la mise à jour de l'entrée d'index secondaire », « impossible de purger un enregistrement » et « tentative de purge de l'entrée d'index secondaire non marquée pour la suppression ». (Bogues n° 19138298, 70214, 21126772 et 21065746)<br>• Le calcul incorrect de la taille de la pile de l'opération qsort entraîne un dépassement de la pile. (Bogue n° 73979)<br>• Enregistrement introuvable dans un index à la restauration. (Bogues n° 70214 et 72419)<br>• ALTER TABLE ajoute la colonne TIMESTAMP sur la mise à jour CURRENT_TIMESTAMP insère des données ZERO (Bogue n° 17392) |
| Mises à jour du moteur de base de données Aurora MySQL : 23/02/2017 (version 1.11) (obsolète) | 1.11.0 | • L'exécution d'une clé étrangère DROP de table ALTER simultanément avec une autre opération DROP fait disparaître la table. (Bogue n° 16095573)<br>• Certaines requêtes INFORMATION_SCHEMA qui utilisaient ORDER BY n'utilisaient pas d'optimisation filesort comme elles le faisaient auparavant. (Bogue n° 16423536)<br>• FOUND_ROWS () renvoie le mauvais nombre de lignes sur une table. (Bogue n° 68458)<br>• Le serveur échoue au lieu d'indiquer une erreur lorsque trop de tables temporaires sont ouvertes. (Bogue n° 18948649) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL : 14/12/2016 (version 1.10) (obsolète) | 1.10.0 | • L'UNION de tables dérivées retourne des résultats erronés avec des clauses '1=0/false'. (Bogue n° 69471)<br>• Le serveur se bloque dans ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS à la 2e exécution de la procédure stockée. (Bogue n° 20755389)<br>• Empêchez les requêtes MySQL de caler trop longtemps au cours de la synchronisation du cache FTS sur le disque en déchargeant la tâche de synchronisation du cache sur un thread séparé, dès que la taille du cache dépasse 10 % de la taille totale. (Bogues n° 22516559, n° 73816) |
| Mises à jour du moteur de base de données Aurora MySQL : 26/10/2016 (version 1.8.1) (obsolète) | 1.8.1 | • OpenSSL a modifié les paramètres de longueur de clé Diffie-Hellman en raison de ce problème. LogJam (Bogue n° 18367167) |
| Mises à jour du moteur de base de données Aurora MySQL : 18/10/2016 (version 1.8) (obsolète) | 1.8.0 | • Lors de l'abandon de tous les index sur une colonne à index multiples, InnoDB ne parvenait pas à bloquer une opération INDEX DEPOSER lorsqu'une contrainte de clé étrangère avait besoin d'un index. (Bogue n° 16896810)<br>• Résoudre l'incident lié à l'ajout d'une contrainte de clé étrangère. (Bogue n° 16413976)<br>• Correction d'un incident se produisant lors de la récupération d'un curseur dans une procédure stockée pendant l'analyse ou du vidage de la table. (Bogue n° 18158639)<br>• Correction d'un bogue d'incrémentation automatique se produisant lorsqu'un utilisateur modifie une table pour changer la valeur AUTO_INCREMENT à moins de la valeur maximale de la colonne d'incrémentation automatique. (Bogue n° 16310273) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL : 30/08/2016 (version 1.7.0) (obsolète) | 1.7.0 | • Amélioration de l'évolutivité par partitionnement du verrou LOCK_grant. (Port WL #8355)<br>• L'ouverture du curseur sur SELECT dans la procédure stockée entraîne une erreur de segmentation. (Bogue de port n° 16499751)<br>• MySQL donne un faux résultat dans certains cas d'utilisation. (Bogue n° 11751794)<br>• Défaillance générale dans GET_SEL_ARG_FOR_KEYPART – causée par le correctif du bogue 11751794. (Bogue n° 16208709)<br>• Mauvais résultats pour une requête simple avec GROUP BY. (Bogue n° 17909656)<br>• Lignes supplémentaires sur requête semi-jointure avec prédicats de plage. (Bogue n° 16221623)<br>• L'ajout d'une clause ORDER BY à la suite d'une sous-requête IN pourrait entraîner le renvoi de lignes en double. (Bogue n° 16308085)<br>• Défaillance générale pour une requête avec balayage large pour GROUP BY, MySQL. (Bogue n° 16222245)<br>• L'analyse d'index large avec prédicat d'entier à quota renvoie des données aléatoires. (Bogue n° 16394084)<br>• Si l'optimiseur utilisait une analyse d'index large, le serveur pouvait quitter pendant la tentative de création d'une table temporaire. (Bogue n° 16436567)<br>• Alors qu'elle ne devrait pas le faire, la fonction COUNT(DISTINCT) comptait les valeurs NULL lorsque l'optimiseur utilisait l'analyse d'index large. (Bogue n° 17222452)<br>• Si une requête utilisait à la fois la fonction MIN()/MAX() et la fonction aggregate_function(DISTINCT) (par exemple, SUM(DISTINCT)) et était exécutée avec l'analyse d'index |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | large, les valeurs produites par la fonction MIN()/MAX() étaient incorrectement définies. (Bogue n° 17217128) |
| Mises à jour du moteur de base de données Aurora MySQL : 01/06/2016 (version 1.6.5) (obsolète) | 1.6.5 | • SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285) |
| Mises à jour du moteur de base de données Aurora MySQL : 06/04/2016 (version 1.6) (obsolète) | 1.6.0 | • BACKPORT Bug #18694052 FIX FOR ASSERTION `!M_ORDERED_REC_BUFFER' FAILED TO 5.6 (Port Bug #18305270)<br>• SEGV IN MEMCPY(), HA_PARTITION::POSITION (Port Bug # 18383840)<br>• WRONG RESULTS WITH PARTITIONING,INDEX_MERGE AND NO PK (Port Bug # 18167648)<br>• FLUSH TABLES FOR EXPORT: ASSERTION IN HA_PARTITION::EXTRA (Port Bug # 16943907)<br>• SERVER CRASH IN VIRTUAL HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST (Port Bug # 16164031)<br>• RANGE OPTIMIZER CRASHES IN SEL_ARG::RB_INSERT() (Port Bug # 16241773) |
| Mises à jour du moteur de base de données Aurora MySQL : 11/01/2016 (version 1.5) (obsolète) | 1.5.0 | • Traitement d'un correctif incomplet dans la recherche en texte intégral MySQL affectant les tables dans lesquelles le nom de base de données commence par un chiffre. (Bogue de port n° 17607956) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL : 03/12/2015 (version 1.4) (obsolète) | 1.4 | • SEGV dans FTSPARSE(). (Bogue n° 16446108)<br>• Le dictionnaire de données InnoDB n'est pas mis à jour lorsque la colonne est renommée. (Bogue n° 19465984)<br>• Incident FTS après qu'une table a été renommée dans une autre base de données. (Bogue n° 16834860)<br>• L'échec de la préparation du déclenchement sur les tables tronquées entraîne l'erreur 1054. (Bogue n° 18596756)<br>• Des modifications des métadonnées peuvent entraîner des problèmes de déclenchement d'exécution. (Bogue n° 18684393)<br>• La matérialisation n'est pas choisie pour le champ UTF8 VARCHAR long. (Bogue n° 17566396)<br>• Plan d'exécution médiocre pour ORDER BY avec la limite X. (Bogue n°16697792)<br>• Bogue de rétroportage n° 11765744 pour 5.1, 5.5 et 5.6. (Bogue n° 17083851)<br>• Problème de mutex dans SQL/SQL_SHOW.CC entraînant. SIG6 Source probable FILL_VARIABLES. (Bogue n° 20788853)<br>• Bogue de rétroportage n° 18008907 pour les versions 5.5+. (Bogue n° 18903155)<br>• Adaptation d'un correctif pour une erreur de dépassement de capacité de pile dans MySQL 5.7. (Bogue n° 19678930) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| Mises à jour du moteur de base de données Aurora MySQL : 16/10/2015 (versions 1.2, 1.3) (obsolète) | 1.2, 1.3 | • L'arrêt d'une requête dans innodb entraîne finalement un incident avec une assertion. (Bogue n° 1608883)<br>• Pour un échec lors de la création d'un nouveau thread pour le planificateur d'événement, l'exécution d'un événement ou une nouvelle connexion, aucun message n'a été consigné dans le journal des erreurs. (Bogue n° 16865959)<br>• Si une connexion a changé la base de données par défaut alors que simultanément une autre connexion a exécuté SHOW PROCESSLIST, la seconde connexion a pu accéder à de la mémoire non valide lors de la tentative d'affichage de la mémoire de base de données par défaut de la première connexion. (Bogue n° 11765252)<br>• PURGE BINARY LOGS, par conception, ne supprime pas les fichiers journaux binaires qui sont utilisés ou actifs, mais n'a pas fourni aucun avis lorsque cela s'est produit. (Bogue n° 13727933)<br>• Pour certaines instructions, des fuites de mémoire pouvaient être générées lorsque l'optimiseur supprimait des clauses de sous-requête superflues. (Bogue n° 15875919)<br>• Pendant l'arrêt, le serveur pouvait essayer de verrouiller un mutex non initialisé. (Bogue n° 16016493)<br>• Une instruction préparée qui utilisait GROUP_CONCAT() et une clause ORDER BY qui nommait plusieurs colonnes pouvait entraîner l'arrêt du serveur. (Bogue n° 16075310)<br>• L'instrumentation du schéma de performance était manquante pour les threads de travailleur de réplicas. (Bogue n° 16083949)<br>• `STOP SLAVE` pouvait provoquer un blocage lorsqu'il était émis en même temps qu'une instruction telle que SHOW STATUS ayant extrait les valeurs pour une ou plusieurs des variables d'état `Slave_retried_transactions` , `Slave_hea` |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
|---|---|---|
| | | `rtbeat_period` , `Slave_received_heartbeats` , `Slave_last_heartbeat` ou `Slave_running` . (Bogue n° 16088188)<br>• Une requête en texte intégral utilisant le mode booléen pouvait ne retourner aucun résultat dans certains cas où le terme recherché était une phrase entre guillemets. (Bogue n° 16206253)<br>• La tentative de l'optimiseur de supprimer les clauses de sous-requête redondantes générait une assertion lors de l'exécution d'une instruction préparée avec une sous-requête dans la clause ON d'une jointure dans une sous-requête. (Bogue n° 16318585)<br>• GROUP_CONCAT instable, incident dans ITEM_SUM::CLEAN_UP_AFTER_REMOVAL. (Bogue n° 16347450)<br>• Une tentative de remplacement de la liste de mots vides de recherche en texte intégral InnoDB par défaut en créant une table InnoDB avec la même structure que INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD générait une erreur. (Bogue n° 16373868)<br>• Après que le thread client sur un travailleur a effectué une opération FLUSH TABLES WITH READ LOCK suivie de quelques mises à jour sur le maître, le travailleur restait bloqué lors de l'exécution de `SHOW SLAVE STATUS`. (Bogue n° 16387720)<br>• Lors de l'analyse d'une chaîne de recherche délimitée telle que « abc-def » dans une recherche en texte intégral, InnoDB utilise à présent les mêmes délimiteurs de mot que MyISAM. (Bogue n° 16419661)<br>• Incident dans FTS_AST_TERM_SET_WILDCARD. (Bogue n° 16429306) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| | | • SEGFAULT dans FTS_AST_VISIT() pour le test RQG de recherche en texte intégral. (Bogue n° 16435855)<br>• Pour les versions de débogage, lorsque l'optimiseur supprimait un Item_ref pointant sur une sous-requête, il entraînait un arrêt du serveur. (Bogue n° 16509874)<br>• La recherche en texte intégral sur des tables InnoDB échouait lors de recherches d'expressions littérales combinées aux opérateurs + ou -. (Bogue n° 16516193)<br>• `START SLAVE`a échoué lorsque le serveur a été démarré avec les options `--master-info-repository=TABLE` relay-log-info-repository =TABLE et avec autocommit défini sur 0, ainsi que. `--skip-slave-start` (Bogue n° 16533802)<br>• Les résultats d'une recherche en texte intégral InnoDB très étendue pouvaient consommer une quantité excessive de mémoire. (Bogue n° 16625973)<br>• Dans les versions de débogage, une assertion pouvait survenir dans OPT_CHECK_ORDER_BY lors de l'utilisation de données binaires directement dans une chaîne de recherche, car les données binaires peuvent inclure des octets NULL et d'autres caractères non significatifs. (Bogue n° 16766016)<br>• Pour certaines instructions, des fuites de mémoire pouvaient être générées lorsque l'optimiseur supprimait des clauses de sous-requête superflues. (Bogue n° 16807641)<br>• Il était possible de générer un interblocage après l'émission de FLUSH TABLES WITH READ LOCK en émettant `STOP SLAVE` dans une nouvelle connexion à l'esclave, puis en émettant `SHOW SLAVE STATUS` à l'aide de la connexion d'origine. (Bogue n° 16856735) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| | | • GROUP_CONCAT() avec un séparateur non valide pouvait causer un arrêt du serveur. (Bogue n° 16870783)<br>• Le serveur a effectué un verrouillage excessif sur les mutex LOCK_active_mi et active_mi->rli->data_lock pour toute instruction SHOW STATUS LIKE 'modèle', même lorsque le modèle ne correspond pas aux variables d'état qui utilisent ces mutex (`Slave_heartbeat_period` , `Slave_last_heartbeat` , `Slave_received_heartbeats` , `Slave_retried_transactions` , `Slave_running` ). (Bogue n° 16904035)<br>• Une recherche en texte intégral utilisant le modificateur IN BOOLEAN MODE entraînait un échec d'assertion. (Bogue n° 16927092)<br>• Une recherche en texte intégral sur les tables InnoDB échouait pour les recherches qui utilisaient l'opérateur booléen +. (Bogue n° 17280122)<br>• Interblocage quadridirectionnel : zombies, purge des journaux binaires, show processlist, show binlogs. (Bogue n° 17283409)<br>• Lorsqu'un thread SQL qui attendait un verrou de validation était supprimé et redémarré, une transaction était ignorée sur le travailleur. (Bogue n° 17450876)<br>• Un échec de recherche en texte intégral InnoDB survenait en raison d'un jeton « non terminé ». La chaîne et la longueur de chaîne devaient être transmises pour la comparaison de chaîne. (Bogue n° 17659310)<br>• Un grand nombre de tables InnoDB partitionnées pouvaient consommer beaucoup plus de mémoire lorsqu'elles étaient utilisées dans MySQL 5.6 ou 5.7 que la mémoire utilisée par les mêmes tables dans les versions précédentes du serveur MySQL. (Bogue n° 17780517) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| | | • Pour les requêtes en texte intégral, ne pas vérifier si num_token était inférieur à max_proximity_item pouvait entraîner une assertion. (Bogue n° 18233051)<br>• Certaines requêtes relatives aux tables INFORMATI ON_SCHEMA TABLES et COLUMNS pouvaient entraîner une utilisation excessive de mémoire lorsque de nombreuses tables InnoDB étaient vides. (Bogue n° 18592390)<br>• Lors de la validation d'une transaction, un indicateur est désormais utilisé pour vérifier si un thread a été créé, plutôt que la vérification du thread lui-même, qui utilise plus de ressources, en particulier lors de l'exécution du serveur avec master_info_repository=TABLE. (Bogue n° 18684222)<br>• Si un thread client sur un travailleur exécutait FLUSH TABLES WITH READ LOCK alors que le maître exécutait une commande DML, l'exécution de `SHOW SLAVE STATUS` dans le même client se bloquait, entraînant un interblocage. (Bogue n° 19843808)<br>• Le tri selon un résultat GROUP_CONCAT() pouvait causer l'arrêt du serveur. (Bogue n° 19880368) |

| Mise à jour du moteur de base de données | Version | Bogues MySQL corrigés |
| --- | --- | --- |
| Mises à jour du moteur de base de données Aurora MySQL : 24/08/2015 (version 1.1) (obsolète) | 1.1 | • Les bases de données InnoDB dont les noms commencent par un chiffre entraînent une erreur d'analyse de la recherche de texte intégral (FTS). (Bogue n° 17607956)<br>• Les recherches de texte intégral InnoDB échouent dans les bases de données dont les noms commencent par un chiffre. (Bogue n° 17161372)<br>• Pour les bases de données InnoDB sous Windows, l'ID d'objet de la recherche de texte intégral (FTS) n'est pas au format hexadécimal attendu. (Bogue n° 16559254)<br>• Une régression de code présentée dans MySQL 5.6 a impacté de manière négative les performances DROP TABLE et ALTER TABLE. Cela pourrait entraîner une baisse des performances entre MySQL Server 5.5.x et 5.6.x. (Bogue n° 16864741) |

# Vulnérabilités de sécurité corrigées dans Aurora MySQL

Vulnérabilités et expositions courantes (CVE) est une liste d'entrées pour les vulnérabilités de cybersécurité connues du public. Chaque entrée contient un numéro d'identification, une description et au moins une référence publique.

Vous trouverez sur cette page une liste des vulnérabilités de sécurité corrigées dans Aurora MySQL. Pour obtenir des informations générales sur la sécurité d'Aurora, consultez Sécurité dans Amazon Aurora dans le Guide de l'utilisateur Amazon Aurora. Pour plus d'informations sur la sécurité d'Aurora MySQL, consultez Sécurité avec Amazon Aurora MySQL dans le Guide de l'utilisateur Amazon Aurora.

Nous vous recommandons de toujours mettre à niveau vers la dernière version d'Aurora pour être protégé contre les vulnérabilités connues. Vous pouvez utiliser cette page pour vérifier si une version particulière d'Aurora MySQL contient un correctif pour une vulnérabilité de sécurité spécifique. Si votre cluster ne possède pas le correctif de sécurité, vous pouvez voir quelle version d'Aurora MySQL vous devriez installer pour ce correctif.

CVEs corrigés dans les versions 1, 2 et 3 d'Aurora MySQL, sont également répertoriés dans les notes de publication de cette version :

- Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 3
- Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 2.
- Mises à jour du moteur de base de données pour Amazon Aurora MySQL version 1 (obsolète)

> **Note**
>
> La version initiale de la version 3 d'Aurora MySQL inclut toutes les corrections apportées à CVEs la version 8.0.23 de la communauté MySQL. Pour CVEs les futurs correctifs, recherchez-les ici et dans les notes de publication de la version 3 d'Aurora MySQL.

CVEs et versions minimales fixes d'Aurora MySQL

- CVE-2024-37371 : 3,08,0
- CVE-2024-25062 : 3,08,0, 2,12,4

- CVE-2024-21185 : 3,08,0
- CVE-2024-21179 : 3,08,0
- CVE-2024-21177 : 3,08,0, 2,12,4
- CVE-2024-21173 : 3,08,0
- CVE-2024-21166 : 3,08,0
- CVE-2024-21165 : 3,08,0
- CVE-2024-21163 : 3,08,0
- CVE-2024-21162 : 3,08,0
- CVE-2024-21160 : 3,08,0
- CVE-2024-21159 : 3,08,0
- CVE-2024-21157 : 3,08,0
- CVE-2024-21142 : 3,08,0, 2,12,4
- CVE-2024-21137 : 3,08,0
- CVE-2024-21135 : 3,08,0
- CVE-2024-21134 : 3,08,0
- CVE-2024-21130 : 3,08,0
- CVE-2024-21129 : 3,08,0
- CVE-2024-21127 : 3,08,0
- CVE-2024-21125 : 3,08,0
- CVE-2024-21097 : 2,12.3, 2,11.6
- CVE-2024-21096 : 2,12.3, 2,11.6
- CVE-2024-21069 : 2,12.3, 2,11.6
- CVE-2024-21062 : 2,12.3, 2,11.6
- CVE-2024-21057 : 2,12.3, 2,11.6
- CVE-2024-21055 : 2,12.3, 2,11.6
- CVE-2024-21054 : 2,12.3, 2,11.6
- CVE-2024-21047 : 2,12.3, 2,11.6
- CVE-2024-21013 : 3,08.0, 2,12.3, 2,11.6
- CVE-2024-21009 : 2,12.3, 2,11.6
- CVE-2024-21008 : 2,12.3, 2,11.6

- CVE-2024-21000 : 3,08,0
- CVE-2024-20998 : 2,12.3, 2,11.6
- CVE-2024-20996 : 3,08,0
- CVE-2024-20993 : 2,12.3, 2,11.6
- CVE-2024-20963 : 2.12.2, 2.11.6, 2,11.5
- CVE-2024-0853 : 3,08.0, 3,06.1, 3,04.3, 2,12.3, 2,11.6
- CVE-2023-44487 : 3,08.0, 3,06.1, 2,12.4, 2,12.3, 2,11.6
- CVE-2023-39975 : 3,07.0, 3,006.0, 3,05.2, 3,04.2, 2.12.2, 2,11.6, 2,11.5
- CVE-2023-38546 : 3,07,0, 3,06,0, 3,04,2, 2,11,5
- CVE-2023-38545 : 3,07.0, 3,006.0, 3,05.2, 3,05.1, 3,050.0.1, 3.04.2, 3.03.3, 2.12.2, 2,12.1, 2.12.0.1, 2,11.6, 2,11.5
- CVE-2023-22084 : 2,11,5
- CVE-2023-22053 : 2.12.1
- CVE-2023-22028 : 2,12.1, 2,11.5
- CVE-2023-22026 : 2,12.1, 2,11.5
- CVE-2023-22015 : 2,12.1, 2,11.5
- CVE-2023-21963 : 3.04.0, 3.03.2, 2.12.0, 2.11.3
- CVE-2023-21912 : 3,04,0, 3,03,2, 2,12,3, 2,12,0, 2,11,6, 2,11.3
- CVE-2023-21840 : 2.12.0
- CVE-2023-0215 : 3.04.0, 3.03.2, 2.12.0, 2.11.3
- CVE-2022-43551 : 3.04.0, 3.03.2, 2.12.0, 2.11.3
- CVE-2022-37434 : 3.05.0, 3.04.0, 3.03.2, 2.12.0, 2.11.3
- CVE-2022-32221 : 3.03.0, 2.12.0, 2.11.1, 2.07.9
- CVE-2022-24407 : 2.12.1, 2.11.4
- CVE-2022-21635 : 3.04.0
- CVE-2022-21556 : 3.04.0
- CVE-2022-21460 : 2.11.0
- CVE-2022-21451 : 3.03.0, 3.02.2, 2.11.0
- CVE-2022-21444 : 3.03.0, 3.02.2, 2.11.0, 2.10.3
- CVE-2022-21417 : 2.11.0
- CVE-2022-21352 : 3.04.0

- CVE-2022-21344 : 2.10.3
- CVE-2022-21304 : 2.11.0, 2.10.3
- CVE-2022-21303 : 2.11.0
- CVE-2022-21245 : 2.11.0, 2.10.3, 2.07.8
- CVE-2022-0778 : 3.02.1, 2.11.0
- CVE-2021-36222 : 3.02.2, 3.01.1, 2.12.0, 2.11.1, 2.11.0, 2.10.3, 2.10.2, 2.07.8
- CVE-2021-35630 : 3.04.0
- CVE-2021-35624 : 3.04.0, 2.10.2
- CVE-2021-35604 : 2.10.2
- CVE-2021-28196 : 2.11.0
- CVE-2021-23841 : 2.11.0, 2.10.0, 2.09.3, 1.23.3
- CVE-2021-22946 : 3.02.0, 3.01.1, 2.12.0
- CVE-2021-22926 : 3.02.2, 3.01.1, 2.11.1, 2.11.0, 2.10.3, 2.10.22.07.8
- CVE-2021-3712 : 2.09.3
- CVE-2021-3449 : 2.11.0, 2.10.0, 2.09.3, 1.23.3
- CVE-2021-2390 : 2.10.2
- CVE-2021-2389 : 2.10.2
- CVE-2021-2385 : 2.10.2
- CVE-2021-2356 : 2.10.2
- CVE-2021-2307 : 2.11.0, 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2226 : 2.11.0, 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2202 : 2.11.0
- CVE-2021-2194 : 2.11.0, 2.10.1
- CVE-2021-2179 : 2.11.0
- CVE-2021-2178 : 2.11.0
- CVE-2021-2174 : 2.11.0, 2.10.1, 2.09.3
- CVE-2021-2171 : 2.11.0, 2.10.1, 2.09.3
- CVE-2021-2169 : 2.12.0, 2.11.1, 2.11.0, 2.10.1, 2.09.3
- CVE-2021-2166 : 2.11.0, 2.10.1, 2.09.3
- CVE-2021-2160 : 2.11.0, 2.10.1, 1.23.4

- CVE-2021-2154 : 2.11.0, 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2144 : 2.07.3
- CVE-2021-2060 : 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2032 : 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2001 : 2.10.1, 2.09.3, 1.23.4
- CVE-2020-28196 : 2.10.0, 2.09.3, 1.23.3
- CVE-2020-14867 : 1.23.2, 1.22.4
- CVE-2020-14812 : 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14793 : 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14790 : 2.10.0, 2.09.2, 2.07.4
- CVE-2020-14776 : 2.10.0
- CVE-2020-14775 : 2.09.2, 2.07.4
- CVE-2020-14769 : 2.09.3, 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14765 : 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14760 : 2.09.2, 2.07.4
- CVE-2020-14672 : 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14567 : 2.10.0, 2.09.1, 2.08.3, 2.07.3
- CVE-2020-14559 : 2.10.0, 2.09.1, 2.08.3, 2.07.3, 1.23.1, 1.22.3
- CVE-2020-14553 : 2.10.0, 2.09.1, 2.08.3, 2.07.3
- CVE-2020-14547 : 2.10.0, 2.09.1, 2.08.3, 2.07.3
- CVE-2020-14540 : 2.10.0, 2.09.1, 2.08.3, 2.07.3
- CVE-2020-14539 : 2.10.0, 1.23.1, 1.22.3
- CVE-2020-11105 : 3,07.0, 3,006.0, 3,05.2, 3.04.2, 2,12.1, 2,11.5
- CVE-2020-11104 : 3,07.0, 3,006.0, 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

# Historique des notes de mise à jour pour Aurora MySQL

Le tableau suivant décrit les versions de la documentation des notes de mise à jour pour Aurora MySQL.

| Modification | Description | Date |
| --- | --- | --- |
| Aurora MySQL version 3.10.2, compatible avec MySQL 8.0.42 | La version 3.10.2 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.42. Pour plus d'informations, consultez les mises à jour du moteur de base de données Aurora MySQL 2025-11-20 (version 3.10.2, compatible avec MySQL 8.0.42). | 20 novembre 2025 |
| Aurora MySQL version 3.11.0, compatible avec MySQL 8.0.43 | La version 3.11.0 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.43. Pour plus d'informations, consultez les mises à jour du moteur de base de données Aurora MySQL 2025-11-13 (version 3.11.0, compatible avec MySQL 8.0.43). | 13 novembre 2025 |
| Aurora MySQL version 3.10.1, compatible avec MySQL 8.0.42 | La version 3.10.1 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.42. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 30 septembre 2025 |

| | | |
|---|---|---|
| Aurora MySQL version 3.10.0, compatible avec MySQL 8.0.42 | La version 3.10.0 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.42. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 31 juillet 2025 |
| Aurora MySQL version 3.09.0, compatible avec MySQL 8.0.40 | La version 3.09.0 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.40. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 14 mai 2025 |
| Aurora MySQL version 3.04.4, compatible avec MySQL 8.0.28 | La version 3.04.4 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.28. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 5 mai 2025 |
| Aurora MySQL version 2.12.5, compatible avec MySQL 5.7.44 | La version 2.12.5 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 5.7.44. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 9 avril 2025 |

| | | |
|---|---|---|
| Aurora MySQL version 3.08.2, compatible avec MySQL 8.0.39 | La version 3.08.2 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.39. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 17 février 2025 |
| Aurora MySQL version 3.08.1, compatible avec MySQL 8.0.39 | La version 3.08.1 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.39. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 17 février 2025 |
| Aurora MySQL version 3.08.0, compatible avec MySQL 8.0.39 | La version 3.08.0 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.39. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 18 novembre 2024 |
| Aurora MySQL version 2.12.4, compatible avec MySQL 5.7.44 | La version 2.12.4 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 5.7.44. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 23 octobre 2024 |

| | | |
|---|---|---|
| Aurora MySQL version 3.07.1, compatible avec MySQL 8.0.36 | La version 3.07.1 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.36. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 23 juillet 2024 |
| Aurora MySQL version 2.11.6, compatible avec MySQL 5.7.12 | La version 2.11.6 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 19 juillet 2024 |
| Aurora MySQL version 2.12.3, compatible avec MySQL 5.7.44 | La version 2.12.3 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 5.7.44. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 9 juillet 2024 |
| Aurora MySQL version 3.06.1, compatible avec MySQL 8.0.34 | La version 3.06.1 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.34. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 26 juin 2024 |

| | | |
|---|---|---|
| Aurora MySQL version 3.04.3, compatible avec MySQL 8.0.28 | La version 3.04.3 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.28. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 26 juin 2024 |
| Aurora MySQL version 3.07.0, compatible avec MySQL 8.0.36 | La version 3.07.0 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.36. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 4 juin 2024 |
| Aurora MySQL version 2.11.5, compatible avec MySQL 5.7.12 | La version 2.11.5 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 26 mars 2024 |
| Aurora MySQL version 2.12.2, compatible avec MySQL 5.7.44 | La version 2.12.2 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 5.7.44. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 19 mars 2024 |

| | | |
|---|---|---|
| Aurora MySQL version 3.04.2, compatible avec MySQL 8.0.28 | La version 3.04.2 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.28. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 15 mars 2024 |
| Aurora MySQL version 3.06.0, compatible avec MySQL 8.0.34 | La version 3.06.0 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.34. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 07 mars 2024 |
| Aurora MySQL version 3.05.2, compatible avec MySQL 8.0.32 | La version 3.05.2 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.32. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 31 janvier 2024 |
| Aurora MySQL version 2.12.1, compatible avec MySQL 5.7.40 | Aurora MySQL version 2.12.1 est disponible. Cette version est compatible avec MySQL 5.7.40. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 28 décembre 2023 |

| | | |
|---|---|---|
| Aurora MySQL version 3.03.3, compatible avec MySQL 8.0.26 | Aurora MySQL version 3.03.3 est disponible. Cette version est compatible avec MySQL 8.0.26. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 8 décembre 2023 |
| Aurora MySQL version 3.05.1, compatible avec MySQL 8.0.32 | Aurora MySQL version 3.05.1 est disponible. Cette version est compatible avec MySQL 8.0.32. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 21 novembre 2023 |
| Aurora MySQL version 3.04.1, compatible avec MySQL 8.0.28 | Aurora MySQL version 3.04.1 est disponible. Cette version est compatible avec MySQL 8.0.28. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 13 novembre 2023 |
| Aurora MySQL version 3.05.0.1, compatible avec MySQL 8.0.32, Bêta | La version 3.05.0.1 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.32. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 30 octobre 2023 |

| | | |
|---|---|---|
| Aurora MySQL version 3.05.0, compatible avec MySQL 8.0.32 | La version 3.05.0 d'Aurora MySQL est disponible. Cette version est compatible avec MySQL 8.0.32. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 25 octobre 2023 |
| Aurora MySQL version 2.12.0.1, compatible avec MySQL 5.7.40, Bêta | Aurora MySQL version 2.12.0.1 Bêta est disponible. Cette version est compatible avec MySQL 5.7.40. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 25 octobre 2023 |
| Aurora MySQL version 2.11.4, compatible avec MySQL 5.7.12 | Aurora MySQL version 2.11.4 est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 17 octobre 2023 |
| Aurora MySQL version 3.03.2, compatible avec MySQL 8.0.26 | Aurora MySQL version 3.03.2 est disponible. Cette version est compatible avec MySQL 8.0.26. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 29 août 2023 |

| | | |
|---|---|---|
| Aurora MySQL version 2.07.10, compatible avec MySQL 5.7.12 | Aurora MySQL version 2.07.10 est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 15 août 2023 |
| Aurora MySQL version 3.04.0, compatible avec MySQL 8.0.28 | Aurora MySQL version 3.04.0 est disponible. Cette version est compatible avec MySQL 8.0.28. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 31 juillet 2023 |
| Aurora MySQL version 2.12.0, compatible avec MySQL 5.7.40 | Aurora MySQL version 2.12.0 est disponible. Cette version est compatible avec MySQL 5.7.40. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 25 juillet 2023 |
| Aurora MySQL version 2.11.3, compatible avec MySQL 5.7.12 | Aurora MySQL version 2.11.3 est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 9 juin 2023 |

| | | |
|---|---|---|
| Aurora MySQL version 3.03.1, compatible avec MySQL 8.0.26 | Aurora MySQL version 3.03.1 est disponible. Cette version est compatible avec MySQL 8.0.26. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 11 mai 2023 |
| Aurora MySQL version 2.07.9, compatible avec MySQL 5.7.12 | Aurora MySQL version 2.07.9 est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 4 mai 2023 |
| Aurora MySQL version 3.02.3, compatible avec MySQL 8.0.23 | Aurora MySQL version 3.02.3 est disponible. Cette version est compatible avec MySQL 8.0.23. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 17 avril 2023 |
| Aurora MySQL version 2.11.2, compatible avec MySQL 5.7.12 | Aurora MySQL version 2.11.2 est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 24 mars 2023 |

| | | |
|---|---|---|
| Aurora MySQL version 3.03.0, compatible avec MySQL 8.0.26 | Aurora MySQL version 3.03.0 est disponible. Cette version est compatible avec MySQL 8.0.26. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 1er mars 2023 |
| Aurora MySQL version 2.11.1, compatible avec MySQL 5.7.12 | Aurora MySQL version 2.11.1 est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 14 février 2023 |
| Aurora MySQL version 3.02.2, compatible avec MySQL 8.0.23 | Aurora MySQL version 3.02.2 est disponible. Cette version est compatible avec MySQL 8.0.23. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 18 novembre 2022 |
| Aurora MySQL version 2.10.3, compatible avec MySQL 5.7 | Aurora MySQL version 2.10.3 est disponible. Cette version est compatible avec MySQL 5.7. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 1er novembre 2022 |

| | | |
|---|---|---|
| Aurora MySQL version 2.11.0, compatible avec MySQL 5.7.12 | Aurora MySQL version 2.11.0 est disponible. Cette version est compatible avec MySQL 5.7.12. Pour plus d'informations, consultez Aurora MySQL version 2 compatible avec MySQL 5.7. | 25 octobre 2022 |
| Aurora MySQL version 3.02.1, compatible avec MySQL 8.0.23 | Aurora MySQL version 3.02.1 est disponible. Cette version est compatible avec MySQL 8.0.23. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 7 septembre 2022 |
| Mise à niveau sur place pour Aurora Serverless v1 compatible avec MySQL 5.6 | Vous pouvez effectuer une mise à niveau sur place pour un cluster Aurora Serverless v1 compatibl e avec MySQL 5.6 afin de convertir un cluster existant en cluster Aurora Serverless v1 compatible avec MySQL 5.7. Pour plus d'informations, consultez Mises à jour du moteur Aurora MySQL Serverless 5.7 du 18/06/2020 (version 2.07.1) | 16 juin 2022 |

| | | |
|---|---|---|
| Mise à niveau sur place pour Aurora Serverless v1 compatible avec MySQL 5.6 | Vous pouvez effectuer une mise à niveau sur place pour un cluster Aurora Serverless v1 compatibl e avec MySQL 5.6 afin de convertir un cluster existant en cluster Aurora Serverless v1 compatible avec MySQL 5.7. Pour plus d'informations, consultez Mises à jour du moteur Aurora MySQL Serverless 5.7 du 18/06/2020 (version 2.07.1) | 16 juin 2022 |
| Les mises à jour du moteur de base de données Aurora MySQL du 16/06/2022 (version 2.07.8) sont disponibl es. | Aurora MySQL version 2.07.8 est disponible. | 16 juin 2022 |
| Aurora MySQL version 3.02.0, compatible avec MySQL 8.0.23 | Aurora MySQL version 3.02.0 est disponible. Cette version est compatible avec MySQL 8.0.23. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 20 avril 2022 |
| Aurora MySQL version 3.01.1, compatible avec MySQL 8.0.23 | Aurora MySQL version 3.01.1 est disponible. Cette version est compatible avec MySQL 8.0.23. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 15 avril 2022 |

| | | |
|---|---|---|
| Première version | Publication initiale des notes de mise à jour pour Aurora MySQL. | 22 mars 2022 |
| Aurora MySQL version 2.10.2 | Aurora MySQL version 2.10.2 est disponible. | 26 janvier 2022 |
| Aurora MySQL version 2.08.4 | Aurora MySQL version 2.08.4 est disponible. | 6 janvier 2022 |
| Aurora MySQL version 2.07.7 | Aurora MySQL version 2.07.7 est disponible. | 24 novembre 2021 |
| Aurora MySQL version 3.01.0, compatible avec MySQL 8.0.23 | Aurora MySQL version 3.01.0 est disponible. Cette version est compatible avec MySQL 8.0.23. Pour plus de détails, consultez Aurora MySQL version 3 compatible avec MySQL 8.0. | 18 novembre 2021 |
| Aurora MySQL version 2.09.3 | Aurora MySQL version 2.09.3 est disponible. | 12 novembre 2021 |
| Aurora MySQL version 2.10.1 | Aurora MySQL version 2.10.1 est disponible. | 21 octobre 2021 |
| Aurora MySQL version 1.23.4 | La version 1.23.4 de Aurora MySQL est disponible. | 30 septembre 2021 |
| Aurora MySQL version 2.07.6 | La version 2.07.6 de Aurora MySQL est disponible. | 2 septembre 2021 |
| Aurora MySQL version 2.07.5 | Aurora MySQL version 2.07.5 est disponible. | 6 juillet 2021 |
| Aurora MySQL version 1.23.3 | Aurora MySQL version 1.23.3 est disponible. | 28 juin 2021 |

| | | |
|---|---|---|
| Aurora MySQL version 1.22.5 | Aurora MySQL version 1.22.5 est disponible. | 3 juin 2021 |
| Aurora MySQL version 2.10.0 | Aurora MySQL version 2.10.0 est disponible. Parmi les points forts, citons la disponibilité accrue des instances de lecteur lors des redémarrages de l'enregistreur, les améliorations apportées aux correctifs sans interruption de service (ZDP), les améliorations apportées au redémarrage sans interruption de service (ZDR) et l'optimisation du cache des journaux binaires. I/O | 25 mai 2021 |
| Aurora MySQL version 1.23.2 | La version 1.23.2 de Aurora MySQL est disponible. | 18 mars 2021 |
| Aurora MySQL version 2.07.4 | La version 2.07.4 de Aurora MySQL est disponible. | 4 mars 2021 |
| Aurora MySQL version 1.22.4 | Aurora MySQL version 1.22.4 est disponible. | 4 mars 2021 |
| Aurora MySQL version 2.09.2 | La version 2.09.2 de Aurora MySQL est disponible. | 26 février 2021 |
| Aurora MySQL version 2.09.1 | Aurora MySQL version 2.09.1 est disponible. | 11 décembre 2020 |
| Aurora MySQL version 1.23.1 | Aurora MySQL version 1.23.1 est disponible. | 24 novembre 2020 |
| Aurora MySQL version 2.08.3 | Aurora MySQL version 2.08.3 est disponible. | 12 novembre 2020 |

| | | |
|---|---|---|
| Aurora MySQL version 2.07.3 | Aurora MySQL version 2.07.3 est disponible. | 10 novembre 2020 |
| Aurora MySQL version 1.22.3 | Aurora MySQL version 1.22.3 est disponible. | 9 novembre 2020 |
| Aurora MySQL version 2.09.0 | Aurora MySQL version 2.09.0 est disponible. | 17 septembre 2020 |
| Aurora MySQL version 1.23.0 | Aurora MySQL version 1.23.0 est disponible. | 2 septembre 2020 |
| Aurora MySQL version 2.08.2 | Aurora MySQL version 2.08.2 est disponible. | 28 août 2020 |
| Aurora MySQL version 2.04.9 | Aurora MySQL version 2.04.9 est disponible. | 14 août 2020 |
| Aurora MySQL version 2.08.1 | Aurora MySQL version 2.08.1 est disponible. | 18 juin 2020 |
| Aurora MySQL version 1.22.2 pour les clusters compatibles avec les requêtes parallèles | Aurora MySQL version 1.22.2 est disponible lorsque vous créez un cluster compatible avec les requêtes parallèles. | 18 juin 2020 |
| Aurora MySQL version 1.20.1 pour les clusters compatibles avec les requêtes parallèles | Aurora MySQL version 1.20.1 est disponible lorsque vous créez un cluster compatible avec les requêtes parallèles. | 11 juin 2020 |
| Aurora MySQL version 2.08.0 | Aurora MySQL version 2.08.0 est disponible. | 2 juin 2020 |
| Aurora MySQL version 1.19.6 pour les clusters compatibles avec les requêtes parallèles | Aurora MySQL version 1.19.6 est disponible lorsque vous créez un cluster compatible avec les requêtes parallèles. | 2 juin 2020 |

| | | |
|---|---|---|
| Aurora MySQL version 2.07.2 | Aurora MySQL version 2.07.2 est disponible. | 17 avril 2020 |
| Aurora MySQL version 1.22.2 | Aurora MySQL version 1.22.2 est disponible. | 5 mars 2020 |
| Aurora MySQL version 1.20.1 | Aurora MySQL version 1.20.1 est disponible. | 5 mars 2020 |
| Aurora MySQL version 1.19.6 | Aurora MySQL version 1.19.6 est disponible. | 5 mars 2020 |
| Aurora MySQL version 1.17.9 | Aurora MySQL version 1.17.9 est disponible. | 5 mars 2020 |
| Aurora MySQL version 2.07.1 | Aurora MySQL version 2.07.1 est disponible. | 23 décembre 2019 |
| Aurora MySQL version 1.22.1 | Aurora MySQL version 1.22.1 est disponible. | 23 décembre 2019 |
| Aurora MySQL version 2.07.0 | Aurora MySQL version 2.07.0 est disponible. | 25 novembre 2019 |
| Aurora MySQL version 1.22.0 | Aurora MySQL version 1.22.0 est disponible. | 25 novembre 2019 |
| Aurora MySQL version 1.21.0 | Aurora MySQL version 1.21.0 est disponible. | 25 novembre 2019 |
| Aurora MySQL version 2.06.0 | Aurora MySQL version 1.21.0 est disponible. | 22 novembre 2019 |
| Aurora MySQL version 2.04.8 | Aurora MySQL version 2.04.8 est disponible. | 20 novembre 2019 |
| Aurora MySQL version 2.04.7 | Aurora MySQL version 2.04.7 est disponible. | 14 novembre 2019 |

| | | |
|---|---|---|
| Aurora MySQL version 2.05.0 | Aurora MySQL version 2.05.0 est disponible. | 11 novembre 2019 |
| Aurora MySQL version 1.20.0 | Aurora MySQL version 1.20.0 est disponible. | 11 novembre 2019 |
| Aurora MySQL version 2.04.6 | Aurora MySQL version 2.04.6 est disponible. | 19 septembre 2019 |
| Aurora MySQL version 1.19.5 | Aurora MySQL version 1.19.5 est disponible. | 19 septembre 2019 |
| Aurora MySQL version 2.04.5 | Aurora MySQL version 2.04.5 est disponible. | 8 juillet 2019 |
| Aurora MySQL version 1.19.2 | Aurora MySQL version 1.19.2 est disponible. | 5 juin 2019 |
| Aurora MySQL version 2.04.4 | Aurora MySQL version 2.04.4 est disponible. | 29 mai 2019 |
| Aurora MySQL version 2.04.3 | Aurora MySQL version 2.04.3 est disponible. | 9 mai 2019 |
| Aurora MySQL version 1.19.1 | Aurora MySQL version 1.19.1 est disponible. | 9 mai 2019 |
| Aurora MySQL version 2.04.2 | Aurora MySQL version 2.04.2 est disponible. | 2 mai 2019 |
| Aurora MySQL version 2.04.1 | Aurora MySQL version 2.04.1 est disponible. | 25 mars 2019 |
| Aurora MySQL version 2.04 | Aurora MySQL version 2.04 est disponible. | 25 mars 2019 |
| Aurora MySQL version 2.03.4 | Aurora MySQL version 2.03.4 est disponible. | 7 février 2019 |

| | | |
|---|---|---|
| Aurora MySQL version 1.19.0 | Aurora MySQL version 1.19.0 est disponible. | 7 février 2019 |
| Aurora MySQL version 2.03.3 | Aurora MySQL version 2.03.3 est disponible. | 18 janvier 2019 |
| Aurora MySQL version 1.17.8 | Aurora MySQL version 1.17.8 est disponible. | 17 janvier 2019 |
| Aurora MySQL version 2.03.2 | Aurora MySQL version 2.03.2 est disponible. | 9 janvier 2019 |
| Aurora MySQL version 2.03.1 | Aurora MySQL version 2.03.1 est disponible. | 24 octobre 2018 |
| Aurora MySQL version 2.03 | Aurora MySQL version 2.03 est disponible. | 11 octobre 2018 |
| Aurora MySQL version 2.02.5 | Aurora MySQL version 2.02.5 est disponible. | 8 octobre 2018 |
| Aurora MySQL version 1.17.7 | Aurora MySQL version 1.17.7 est disponible. | 8 octobre 2018 |
| Aurora MySQL version 2.02.4 | Aurora MySQL version 2.02.4 est disponible. | 21 septembre 2018 |
| Aurora MySQL version 1.18.0 | Aurora MySQL version 1.18.0 est disponible. | 20 septembre 2018 |
| Aurora MySQL version 1.17.6 | Aurora MySQL version 1.17.6 est disponible. | 6 septembre 2018 |

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.