



開發人員指南

Amazon DocumentDB



Amazon DocumentDB: 開發人員指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

| | |
|---------------------------------------|----|
| 什麼是 Amazon DocumentDB | 1 |
| 概觀 | 1 |
| 叢集 | 3 |
| 執行個體 | 3 |
| 區域和 AZ | 5 |
| 區域 | 6 |
| 可用區域 | 6 |
| 定價 | 8 |
| 免費試用 | 8 |
| 監控 | 8 |
| 介面 | 9 |
| AWS Management Console | 9 |
| AWS CLI | 9 |
| mongo 殼層 | 9 |
| MongoDB 驅動程式 | 9 |
| 後續步驟？ | 9 |
| 運作方式 | 10 |
| Amazon DocumentDB 端點 | 11 |
| TLS 支援 | 14 |
| Amazon DocumentDB 存儲 | 14 |
| Amazon DocumentDB 複寫 | 15 |
| Amazon DocumentDB 可靠性 | 15 |
| 讀取偏好設定選項 | 16 |
| TTL 刪除 | 20 |
| 可計費資源 | 20 |
| 什麼是文件資料庫？ | 23 |
| 使用案例 | 23 |
| 了解文件 | 24 |
| 使用文件 | 29 |
| 開始使用指南 | 41 |
| 必要條件 | 41 |
| 步驟 1：建立AWS Cloud9環境 | 43 |
| 步驟 2：建立安全群組 | 44 |
| 第 3 步：創建一個 Amazon DocumentDB 集群 | 46 |

| | |
|--|-----|
| 第 4 步：安裝蒙戈外殼 | 48 |
| 步驟 5：Connect 到您的 Amazon DocumentDB 集群 | 49 |
| 步驟 6：插入和查詢資料 | 51 |
| 步驟 7：探索 | 53 |
| 使用 AWS CloudFormation 的快速入門 | 54 |
| 必要條件 | 54 |
| 所需的 IAM 許可 | 55 |
| Amazon EC2 金鑰對 | 56 |
| 啟動 Amazon DocumentDB 堆棧 AWS CloudFormation | 56 |
| 存取 Amazon DocumentDB 叢集 | 61 |
| 終止保護及刪除保護 | 61 |
| MongoDB 性 | 63 |
| MongoDB 性 | 63 |
| Amazon DocumentDB 5.0 中的新功能 | 63 |
| 開始使用 Amazon DocumentDB 5.0 | 64 |
| 升級或遷移到 Amazon DocumentDB 4.0 | 64 |
| 功能差異 | 64 |
| MongoDB 4.0 兼容性 | 66 |
| Amazon DocumentDB 4.0 功能 | 66 |
| 開始使用 Amazon DocumentDB 4.0 | 67 |
| 升級或遷移到 Amazon DocumentDB 4.0 | 68 |
| 功能差異 | 68 |
| 交易 | 69 |
| 要求 | 69 |
| 最佳實務 | 69 |
| 限制 | 70 |
| 監控與診斷 | 70 |
| 交易隔離層級 | 71 |
| 使用案例 | 72 |
| 多重對帳單交易 | 72 |
| 多重收集異動 | 73 |
| 回呼 API 的交易 API 範例 | 75 |
| 核心 API 的交易 API 範例 | 75 |
| 支援的 命令 | 108 |
| 不支援 | 109 |
| 工作階段 | 110 |

| | |
|---------------------------------|-----|
| 因果一致性 | 110 |
| 可重試寫入 | 111 |
| 交易錯誤 | 111 |
| 最佳實務 | 112 |
| 基本操作準則 | 112 |
| 執行個體大小設定 | 113 |
| 使用索引 | 114 |
| 建立索引 | 114 |
| 索引選擇性 | 115 |
| 索引對編寫數據的影響 | 115 |
| 識別遺失索引 | 115 |
| 識別未使用的索引 | 115 |
| 安全最佳實務 | 116 |
| 成本最佳化 | 116 |
| 使用指標來識別效能問題 | 117 |
| 檢視效能指標 | 117 |
| 設定 CloudWatch 鬧鐘 | 117 |
| 評估效能指標 | 117 |
| 調校查詢 | 119 |
| TTL 和時間序列工作負載 | 119 |
| 遷移 | 119 |
| 使用叢集參數群組 | 120 |
| 彙總管道查詢 | 120 |
| batchInsert 和 batchUpdate | 120 |
| MongoDB 的功能差異 | 121 |
| Amazon DocumentDB 的功能優勢 | 121 |
| 隱含交易 | 121 |
| 更新的功能差異 | 122 |
| 編製陣列索引 | 123 |
| 多金鑰索引 | 123 |
| 字串中的 Null 字元 | 124 |
| 以角色為基礎的存取控制 | 124 |
| \$regex 索引功能 | 124 |
| 巢狀文件的投影 | 125 |
| MongoDB 的功能差異 | 125 |
| \$vectorSearch 運算子 | 126 |

| | |
|---|-----|
| OpCountersCommand | 126 |
| 管理員資料庫和集合 | 126 |
| cursormaxTimeMS | 126 |
| explain() | 127 |
| 欄位名稱限制 | 127 |
| 索引建置 | 128 |
| 在路徑中使用空鍵查找 | 128 |
| MongoDB API、營運和資料類型 | 128 |
| mongodump和mongorestore公用事業 | 128 |
| 結果排序 | 129 |
| 可重試的寫入 | 129 |
| 稀疏索引 | 129 |
| 使用 \$all 表達式內的 \$elemMatch | 130 |
| \$ne、\$nin、\$nor、\$not\$exists、和\$elemMatch索引 | 130 |
| \$lookup | 131 |
| 支援的 MongoDB API、營運和資料類型 | 135 |
| 資料庫命令 | 135 |
| 管理命令 | 136 |
| 聚合 | 137 |
| 身分驗證 | 138 |
| 診斷命令 | 138 |
| 查詢和寫入操作 | 139 |
| 角色管理命令 | 140 |
| 階段作業命 | 141 |
| 使用者管理 | 141 |
| 分片命令 | 142 |
| 查詢和投影運算子 | 144 |
| 陣列運算子 | 144 |
| 位元運算子 | 144 |
| 註解運算子 | 145 |
| 比較運算子 | 145 |
| 元素運算子 | 145 |
| 評估查詢運算子 | 146 |
| 邏輯運算子 | 146 |
| 投影運算子 | 146 |
| 更新運算子 | 147 |

| | |
|---|-----|
| 陣列運算子 | 147 |
| 位元運算子 | 148 |
| 欄位運算子 | 148 |
| 更新修改程式 | 148 |
| Geospatial (地理空間) | 149 |
| 幾何指定元 | 149 |
| 查詢選取器 | 149 |
| 游標方法 | 150 |
| 彙總管道運算子 | 152 |
| 累加器表達式 | 153 |
| 算術運算子 | 153 |
| 陣列運算子 | 154 |
| 布林值運算子 | 155 |
| 比較運算子 | 156 |
| 條件式表達式運算子 | 156 |
| 日期類型運算子 | 156 |
| 資料大小運算子 | 157 |
| 日期運算子 | 157 |
| 常值運算子 | 158 |
| 合併運算子 | 158 |
| 自然操作 | 158 |
| 設定運算子 | 159 |
| 階段運算子 | 159 |
| 字串運算子 | 161 |
| 系統變數 | 162 |
| 文字搜尋運算子 | 162 |
| 類型轉換運算子 | 163 |
| 變數運算子 | 163 |
| 雜項運算子 | 163 |
| 資料類型 | 164 |
| 索引和索引屬性 | 165 |
| 索引 | 165 |
| 索引屬性 | 166 |
| 生成式 AI | 167 |
| SageMaker 帆布 | 167 |
| 如何使用 Canvas 構建無代碼 ML 模型 SageMaker | 167 |

| | |
|--|-----|
| 設定 SageMaker 網域和使用者設定檔 | 168 |
| 設定 Amazon DocumentDB 和畫布的 IAM 存取權限 SageMaker | 168 |
| 為 SageMaker Canvas 建立資料庫使用者和角色 | 168 |
| 可用地區 | 169 |
| 向量搜尋 | 169 |
| 插入向量 | 170 |
| 建立向量索引 | 170 |
| 取得索引定義 | 174 |
| 查詢向量 | 174 |
| 功能和限制 | 177 |
| 最佳實務 | 178 |
| 遷移到 Amazon DocumentDB | 180 |
| 在版本之間移轉 | 180 |
| 步驟 1：啟用變更串流 | 181 |
| 步驟 2：修改變更串流保留持續時間 | 181 |
| 步驟 3：遷移索引 | 182 |
| 步驟 4：建立 AWS DMS 複製執行個體 | 183 |
| 步驟 5：建立 AWS DMS 來源端點 | 185 |
| 步驟 6：建立 AWS DMS 目標端點 | 187 |
| 步驟 7：建立並執行移轉任務 | 189 |
| 步驟 8：將應用程式端點變更為目標 Amazon DocumentDB 叢集 | 191 |
| 遷移工具 | 191 |
| AWS Database Migration Service | 191 |
| 命令列公用程式 | 191 |
| 探索 | 192 |
| 規劃：Amazon DocumentDB 叢集需求 | 195 |
| 遷移方法 | 198 |
| 離線 | 198 |
| 線上 | 199 |
| 混合 | 200 |
| 遷移來源 | 202 |
| 遷移連線能力 | 202 |
| 測試 | 205 |
| 遷移計畫測試注意事項 | 205 |
| 效能測試 | 208 |
| 容錯移轉測試 | 209 |

| | |
|--|-----|
| 其他資源 | 209 |
| 移轉教戰手冊 | 209 |
| 遷移程序 | 209 |
| 其他資源 | 213 |
| 升級 Amazon DocumentDB 引擎版本 | 215 |
| 先決條件和限制 | 215 |
| 就地主要版本升級的最佳作法 | 218 |
| 使用複製的叢集就地測試主要版本升級 | 218 |
| 在就地主要版本升級之前 | 218 |
| 在就地主要版本升級期間 | 219 |
| 在就地主要版本升級之後 | 221 |
| 執行就地主要版本升級 | 222 |
| 疑難排解就地主要版本升級 | 225 |
| Amazon DocumentDB 3.6/4.0 到 5.0 升級的集群和新的 Amazon DocumentDB 5.0 集群之間的 差異 | 225 |
| 安全性 | 226 |
| 資料保護 | 227 |
| 用戶端欄位層級加密 | 228 |
| 加密靜態資料 | 235 |
| 對傳輸中資料進行加密 | 239 |
| 金鑰管理 | 249 |
| 身分和存取權管理 | 249 |
| 物件 | 250 |
| 使用身分驗證 | 250 |
| 使用政策管理存取權 | 253 |
| Amazon DocumentDB 如何與 IAM 一起工作 | 255 |
| 身分型政策範例 | 262 |
| 故障診斷 | 264 |
| 管理您的 Amazon DocumentDB 資源的訪問許可 | 266 |
| 使用身分類型政策 (IAM 政策) | 270 |
| AWS Amazon DocumentDB 的受管政策 | 274 |
| Amazon DocumentDB API 許可參考 | 291 |
| Amazon DocumentDB 資料庫 | 299 |
| 主要和serviceadmin使用者 | 299 |
| 建立其他使用者 | 299 |
| 自動輪換密碼 | 301 |

| | |
|--------------------------------------|-----|
| 以角色為基礎的存取控制 | 302 |
| RBAC 概念 | 303 |
| RBAC 內建角色入門 | 304 |
| 開始使用 RBAC 使用者定義角色 | 308 |
| 以使用者身分連接到 Amazon DocumentDB | 312 |
| 一般命令 | 314 |
| 功能差異 | 319 |
| 限制 | 319 |
| 使用角色型存取控制存取資料庫 | 319 |
| 記錄和監控 | 324 |
| 更新憑證 | 324 |
| 更新您的應用程式和 Amazon DocumentDB 叢集 | 325 |
| 故障診斷 | 328 |
| 常見問答集 | 328 |
| 更新憑證 — GovCloud (美國西部) | 334 |
| 更新您的應用程式和 Amazon DocumentDB 叢集 | 325 |
| 疑難排解 | 328 |
| 常見問答集 | 328 |
| 合規驗證 | 343 |
| 恢復能力 | 344 |
| 基礎設施安全 | 345 |
| 安全最佳實務 | 345 |
| 稽核事件 | 346 |
| 支援的事件 | 347 |
| 啟用稽核 | 350 |
| 停用稽核 | 356 |
| 存取您的稽核活動 | 358 |
| 備份與還原 | 360 |
| 備份與還原：概念 | 361 |
| 了解備份儲存用量 | 362 |
| 傾印、還原、匯入和匯出資料 | 363 |
| mongodump | 363 |
| mongorestore | 364 |
| mongoexport | 365 |
| mongoimport | 365 |
| 教學課程 | 366 |

| | |
|----------------------------------|-----|
| 叢集快照考量 | 368 |
| 備份儲存 | 369 |
| 備份時段 | 369 |
| 備份保留期間 | 370 |
| 複製叢集快照加密 | 370 |
| 比較自動與手動快照 | 371 |
| 建立手動叢集快照 | 372 |
| 複製叢集快照 | 375 |
| 複製共用快照 | 376 |
| 複製快照 AWS 區域 | 376 |
| 限制 | 376 |
| 處理加密 | 377 |
| 參數群組考量 | 377 |
| 複製叢集快照 | 377 |
| 共用叢集快照 | 384 |
| 共用加密快照 | 384 |
| 共享快照 | 387 |
| 從叢集快照還原 | 389 |
| 時間點還原 | 396 |
| 刪除叢集快照 | 401 |
| 管理亞馬遜文檔數據庫 | 403 |
| 營運任務概觀 | 403 |
| 新增複本到 Amazon DocumentDB 叢集 | 404 |
| 管理叢集和執行個體 | 404 |
| 建立叢集快照 | 407 |
| 從快照還原 | 408 |
| 從叢集中移除執行個體 | 409 |
| 刪除叢集 | 409 |
| 全域叢集 | 410 |
| 什麼是全域叢集？ | 410 |
| 全局集群如何有用？ | 410 |
| 全域叢集目前有哪些限制？ | 410 |
| 快速入門指南 | 411 |
| 管理全域叢集 | 425 |
| 連接全域叢集 | 432 |
| 監視全域叢集 | 433 |

| | |
|-----------------------------------|-----|
| 災難復原 | 434 |
| 管理 叢集 | 435 |
| 了解叢集 | 436 |
| 叢集設定 | 438 |
| 叢集儲存配置 | 440 |
| 決定叢集的狀態 | 443 |
| 叢集生命週期 | 444 |
| 擴展叢集 | 481 |
| 複製叢集的磁碟區 | 484 |
| 了解叢集容錯 | 496 |
| 管理執行個體 | 497 |
| 管理執行個體類別 | 497 |
| 決定執行個體的狀態 | 506 |
| 執行個體生命週期 | 506 |
| 管理子網路群組 | 529 |
| 建立子網路群組 | 529 |
| 描述子網路群組 | 534 |
| 修改子網路群組 | 537 |
| 刪除子網路群組 | 540 |
| 高可用性和複寫 | 541 |
| 讀取擴展 | 542 |
| 高可用性 | 542 |
| 新增 複本 | 543 |
| 容錯移轉 | 543 |
| 複寫延遲 | 547 |
| 管理索引 | 548 |
| Amazon DocumentDB 索引創建 | 548 |
| 管理文件壓縮 | 553 |
| 指導方針 | 553 |
| 啟用文件壓縮 | 553 |
| 監控文件壓縮 | 554 |
| 管理現有的集合 | 555 |
| 管理事件 | 555 |
| 檢視事件類別 | 555 |
| 檢視 Amazon DocumentDB 事件事件範例 | 558 |
| 選擇區域與可用區域 | 560 |

| | |
|--------------------------------|-----|
| 區域可用性 | 561 |
| 管理叢集參數群組 | 563 |
| 描述叢集參數群組 | 563 |
| 建立叢集參數群組 | 570 |
| 修改叢集參數群組 | 572 |
| 修改叢集以使用自訂叢集參數群組 | 577 |
| 複製叢集參數群組 | 578 |
| 重設叢集參數群組 | 580 |
| 刪除叢集參數群組 | 582 |
| 叢集參數參考 | 585 |
| 瞭解端點 | 596 |
| 尋找叢集端點 | 597 |
| 搜尋執行個體的端點 | 599 |
| 連接至端點 | 603 |
| 了解 Amazon DocumentDB ARN | 604 |
| 建構 ARN | 605 |
| 尋找 ARN | 608 |
| 標記資源 | 610 |
| 資源標籤概觀 | 610 |
| 標籤限制 | 611 |
| 新增或更新標籤 | 611 |
| 列出標籤 | 613 |
| 移除標籤 | 614 |
| 維護 Amazon DocumentDB | 615 |
| 判斷待定的維護動作 | 616 |
| 決定擱置的維護動作 | 617 |
| 套用引擎更新 | 619 |
| 使用者啟動更新 | 622 |
| 管理您的維護時段 | 623 |
| 作業系統更新 | 625 |
| 了解服務連結角色 | 628 |
| 服務連結角色許可 | 629 |
| 建立服務連結角色 | 630 |
| 修改服務連結角色 | 631 |
| 刪除服務連結角色 | 631 |
| 支援 Amazon 文件資料庫服務連結角色的區域 | 632 |

| | |
|--|-----|
| 使用 Amazon DocumentDB 彈性集群 | 633 |
| 彈性叢集使用案例 | 633 |
| 使用者設定檔 | 634 |
| 內容管理和歷史記錄 | 634 |
| 彈性叢集的優點 | 634 |
| AWS 服務整合 | 634 |
| 區域和版本可用性 | 634 |
| 區域可用性 | 634 |
| 版本可用性 | 635 |
| 限制 | 635 |
| 彈性叢集管理 | 635 |
| 查詢和寫入操作 | 636 |
| 集合和索引管理 | 636 |
| 管理和診斷 | 636 |
| 選擇加入功能 | 637 |
| 運作方式 | 637 |
| Amazon DocumentDB 彈性集群分片 | 637 |
| 彈性叢集移轉 | 641 |
| 彈性叢集擴展 | 641 |
| 彈性群集可靠性 | 641 |
| 彈性叢集儲存與可用性 | 641 |
| Amazon DocumentDB 4.0 和彈性集群之間的功能差異 | 641 |
| 開始使用 | 643 |
| 設定 | 643 |
| 步驟 1：建立彈性叢集 | 645 |
| 步驟 2：建立 AWS Cloud9 環境 | 651 |
| 第 3 步：安裝蒙戈外殼 | 653 |
| 步驟 4：Connect 到新的彈性叢集 | 654 |
| 第 5 步：分片您的收藏；插入和查詢數據 | 654 |
| 最佳實務 | 657 |
| 選擇分片 | 657 |
| 連線管理 | 657 |
| 未分片的集合 | 657 |
| 擴展 | 658 |
| 監控彈性叢集 | 658 |
| 管理彈性叢集 | 659 |

| | |
|--|-----|
| 修改彈性叢集配置 | 659 |
| 監視彈性叢集 | 662 |
| 刪除彈性叢集 | 665 |
| 管理彈性叢集快照 | 667 |
| 停止及啟動彈性叢集 | 680 |
| 靜態資料加密 | 684 |
| Amazon DocumentDB 彈性集群如何使用授權 AWS KMS | 686 |
| 建立客戶受管金鑰受管 | 686 |
| 監控 Amazon DocumentDB Elastic Clusters Clusters 的加密金鑰，以 | 687 |
| 進一步了解 | 693 |
| 服務連結角色 | 693 |
| 彈性叢集的服務連結角色權限 | 693 |
| Amazon DocumentDB | 697 |
| 監視叢集的狀態 | 698 |
| 叢集狀態值 | 698 |
| 監視叢集的狀態 | 699 |
| 監視執行個體的狀態 | 701 |
| 執行個體狀態值 | 702 |
| 使用AWS Management Console或監視執行個體狀態AWS CLI | 703 |
| 執行個體運作狀態值 | 705 |
| 監視執行個體健全狀態AWS Management Console | 706 |
| 檢視 Amazon DocumentDB | 707 |
| 事件訂閱 | 710 |
| 訂閱事件 | 711 |
| 管理訂閱 | 714 |
| 類別和訊息 | 718 |
| 使用監控亞馬遜文檔數據庫 CloudWatch | 721 |
| Amazon DocumentDB 指標 | 721 |
| 檢視 CloudWatch 資料 | 730 |
| 亞馬遜文檔數據庫尺寸 | 736 |
| 監視反應 | 736 |
| 監控資料庫連線 | 736 |
| 使用記錄 Amazon DocumentDB API 呼叫 CloudTrail | 736 |
| Amazon DocumentDB 信息 CloudTrail | 737 |
| 分析操作 | 738 |
| 受支援的 操作 | 738 |

| | |
|---|-----|
| 限制 | 739 |
| 啟用 Profiler | 739 |
| 停用 Profiler | 743 |
| 停用 Profiler 日誌匯出 | 744 |
| 存取您的 Profiler 日誌 | 746 |
| 常用查詢 | 747 |
| 使用績效詳情進行監控 | 747 |
| Performance Insights 概念 | 748 |
| 啟用和停用績效詳情 | 751 |
| 設定績效詳情的存取政策 | 754 |
| 使用績效詳情儀表板來分析指標 | 759 |
| 使用績效詳情 API 來擷取指標 | 776 |
| 亞馬遜 CloudWatch 指標的 Performance Insights | 790 |
| 計數器指標的 Performance Insights | 792 |
| 使用 Amazon DocumentDB 開發 | 794 |
| 以程式設計方式連線 | 794 |
| 判斷 tls 值 | 795 |
| 使用已啟用的 TLS 連線 | 797 |
| 使用已停用的 TLS 連線 | 810 |
| 使用變更串流功能 | 818 |
| 受支援的 操作 | 819 |
| 帳單 | 819 |
| 限制 | 820 |
| 啟用變更串流 | 820 |
| 範例 | 822 |
| 完整文件查閱 | 825 |
| 繼續變更串流 | 825 |
| 使用恢復變更串流startAtOperationTime | 827 |
| 變更串流中的交易 | 829 |
| 修改變更串流日誌保留期間 | 829 |
| 使用AWS Lambda使用變更串流 | 832 |
| 限制 | 832 |
| 使用 JSON 結構定義驗證 | 833 |
| 建立和使用 JSON 結構描述驗證 | 833 |
| 支援關鍵字 | 841 |
| 限制 | 842 |

| | |
|-------------------------------|-----|
| 以複本集形式連接 | 843 |
| 使用叢集連線 | 845 |
| 多個連線集區 | 846 |
| 總結 | 847 |
| 從亞馬遜 VPC 外部連接 | 847 |
| 使 Connect 工作室 3T 連線 | 848 |
| 先決條件 | 848 |
| 與工作室 3T Connect | 849 |
| Connect 使用 DataGrip | 859 |
| 先決條件 | 859 |
| Connect 使用 DataGrip | 860 |
| DataGrip 功能 | 866 |
| 使用 Amazon EC2 Connect | 867 |
| 必要條件 | 867 |
| 自動 Connect Amazon EC2 | 868 |
| 手動 Connect Amazon EC2 | 889 |
| 使 Connect JDBC 驅動程式進行連線 | 904 |
| 入門 | 905 |
| 從桌面 Connect | 906 |
| Connect 來源 DbVisualizer | 909 |
| JDBC 自動產生結構描述 | 912 |
| SQL Support 與限制 | 920 |
| 疑難排解 | 920 |
| 使用 ODBC 驅動程式進行 Connect | 920 |
| 入門 | 921 |
| 在視窗中設定 ODBC 驅動程式 | 922 |
| 從微軟 Connect | 927 |
| Connect 微軟 Power 向版 | 929 |
| 自動生成模式 | 935 |
| SQL 支援和限制 | 936 |
| 疑難排解 | 936 |
| 配額和限制 | 937 |
| 支援的執行個體類型 | 937 |
| 支援地區 | 939 |
| 區域配額 | 940 |
| 彙總限制 | 943 |

| | |
|---------------------------------|-----|
| 叢集限制 | 943 |
| 執行個體限制 | 944 |
| 命名限制 | 946 |
| TTL 限制 | 947 |
| 彈性叢集限制 | 947 |
| 彈性叢集碎片限制 | 948 |
| 每個碎片的彈性叢集 CPU、記憶體、連線和游標限制 | 949 |
| 查詢 | 950 |
| 查詢文件 | 950 |
| 擷取所有文件 | 951 |
| 相符欄位值 | 951 |
| 嵌入式文件 | 951 |
| 內嵌文件中的欄位值 | 951 |
| 匹配數組 | 952 |
| 匹配數組中的值 | 952 |
| 使用運算子 | 952 |
| 查詢計劃 | 952 |
| 查詢計畫 | 952 |
| 查詢計畫快取 | 954 |
| 解釋結果 | 954 |
| 掃描和篩選階段 | 955 |
| 索引交集 | 956 |
| 索引聯集 | 957 |
| 多個索引交點/聯集 | 958 |
| 化合物索引 | 958 |
| 排序階段 | 959 |
| 小組賽階段 | 959 |
| 地理空間資料 | 959 |
| 概要 | 1 |
| 索引和存儲地理空間數據 | 960 |
| 查詢地理空間資料 | 961 |
| 限制 | 965 |
| 部分索引 | 965 |
| 創建一個部分索引 | 965 |
| 支援的運算子 | 966 |
| 使用部分索引進行查詢 | 966 |

| | |
|--|------|
| 部分索引功能 | 967 |
| 部分索引限制 | 971 |
| 文字搜尋 | 972 |
| 支援的功能 | 972 |
| 使用 Amazon DocumentDB 文本索引 | 973 |
| 與 MongoDB 的差異 | 978 |
| 最佳做法和指引 | 978 |
| 限制 | 979 |
| 疑難排解 | 980 |
| 連線問題 | 980 |
| 無法連接到 Amazon DocumentDB 端點 | 980 |
| 測試 Amazon DocumentDB 執行個體的連線 | 985 |
| 連線到無效的端點 | 985 |
| 索引建立 | 986 |
| 索引組建失敗 | 987 |
| 背景索引建置延遲問題和失敗 | 987 |
| 效能與資源使用率 | 987 |
| 檢視插入、更新和刪除統計資料 | 988 |
| 分析快取效能 | 989 |
| 尋找並終止長時間執行或封鎖的查詢 | 990 |
| 查看查詢計畫和最佳化查詢 | 992 |
| 如何在彈性集群中查看查詢計劃？ | 994 |
| 列出執行個體上所有正在執行的操作 | 996 |
| 知道查詢正在進行 | 998 |
| 判斷系統突然執行緩慢的原因 | 1001 |
| 判斷造成高 CPU 使用率的原因 | 1002 |
| 在執行個體上尋找開啟的游標 | 1003 |
| 檢視目前的 Amazon DocumentDB 引擎版本 | 1003 |
| 分析索引使用情況，並識別未使用 | 1004 |
| 識別遺漏的索引 | 1006 |
| 有用的查詢摘要 | 1007 |
| 資源管理 API 參考 | 1009 |
| 動作 | 1009 |
| Amazon DocumentDB (with MongoDB compatibility) | 1012 |
| Amazon DocumentDB Elastic Clusters | 1183 |
| 資料類型 | 1244 |

| | |
|--|------|
| Amazon DocumentDB (with MongoDB compatibility) | 1245 |
| Amazon DocumentDB Elastic Clusters | 1320 |
| 常見錯誤 | 1335 |
| 常見參數 | 1337 |
| 版本備註 | 1340 |
| 2024年4月3日 | 1341 |
| 新功能 | 1342 |
| 錯誤修復和其他更改 | 1342 |
| 2024年2月22日 | 1343 |
| 新功能 | 1343 |
| 2024年1月30日 | 1343 |
| 新功能 | 1343 |
| 2024年1月10日 | 1344 |
| 新功能 | 1344 |
| 錯誤修復和其他更改 | 1345 |
| 2023年12月20日 | 1345 |
| 其他變更 | 1345 |
| 2023年12月13日 | 1345 |
| 新功能 | 1345 |
| 2023年11月29日 | 1345 |
| 新功能 | 1345 |
| 2023年11月21日 | 1345 |
| 新功能 | 1345 |
| 2023年11月17日 | 1346 |
| 新功能 | 1346 |
| 錯誤修復和其他更改 | 1346 |
| 2023年11月6日 | 1346 |
| 新功能 | 1346 |
| 錯誤修復和其他更改 | 1346 |
| 2023年10月20日 | 1347 |
| 其他變更 | 1347 |
| 2023年9月25日 | 1347 |
| 新功能 | 1347 |
| 2023年9月20日 | 1347 |
| 新功能 | 1347 |
| 2023年9月15日 | 1347 |

| | |
|------------------------|------|
| 新功能 | 1347 |
| 2023 年 9 月 11 日 | 1348 |
| 新功能 | 1348 |
| 2023 年 8 月 3 日 | 1348 |
| 新功能 | 1348 |
| 2023 年 7 月 13 日 | 1348 |
| 新功能 | 1348 |
| 錯誤修復和其他更改 | 1349 |
| 2023 年 6 月 7 日 | 1349 |
| 錯誤修復和其他更改 | 1349 |
| 2023 年 5 月 10 日 | 1349 |
| 錯誤修復和其他更改 | 1349 |
| 2023 年 4 月 4 日 | 1349 |
| 錯誤修復和其他更改 | 1349 |
| 2023 年 3 月 22 日 | 1350 |
| 新功能 | 1350 |
| 2023 年 3 月 1 日 | 1350 |
| 新功能 | 1350 |
| 2023 年 2 月 27 日 | 1351 |
| 錯誤修復和其他更改 | 1351 |
| 2023 年 2 月 2 日 | 1351 |
| 錯誤修復和其他更改 | 1351 |
| 2022 年 11 月 30 日 | 1351 |
| 新功能 | 1351 |
| 2022 年 8 月 9 日 | 1352 |
| 新功能 | 1352 |
| 錯誤修復和其他更改 | 1352 |
| 2022 年 7 月 25 日 | 1352 |
| 新功能 | 1352 |
| 2022 年 6 月 27 日 | 1352 |
| 新功能 | 1352 |
| 2022 年 4 月 29 日 | 1353 |
| 新功能 | 1353 |
| 2022 年 4 月 7 日 | 1353 |
| 新功能 | 1353 |
| 2022 年 3 月 16 日 | 1353 |

| | |
|------------------------|-----------|
| 新功能 | 1353 |
| 2022 年 2 月 8 日 | 1353 |
| 新功能 | 1353 |
| 2022 年 1 月 24 日 | 1353 |
| 新功能 | 1353 |
| 2022 年 1 月 21 日 | 1354 |
| 新功能 | 1354 |
| 2021 年 10 月 25 日 | 1354 |
| 新功能 | 1354 |
| 錯誤修復和其他更改 | 1355 |
| 2021 年 6 月 24 日 | 1355 |
| 新功能 | 1355 |
| 2021 年 5 月 4 日 | 1355 |
| 新功能 | 1355 |
| 錯誤修復和其他更改 | 1356 |
| 2021 年 1 月 15 日 | 1356 |
| 新功能 | 1356 |
| 2020 年 11 月 9 日 | 1357 |
| 新功能 | 1357 |
| 錯誤修復和其他更改 | 1358 |
| 2020 年 10 月 30 日 | 1359 |
| 新功能 | 1359 |
| 錯誤修復和其他更改 | 1359 |
| 2020 年 9 月 22 日 | 1359 |
| 新功能 | 1359 |
| 錯誤修復和其他更改 | 1360 |
| 2020 年 7 月 10 日 | 1360 |
| 新功能 | 1360 |
| 錯誤修復和其他更改 | 1360 |
| 2020 年 6 月 30 日 | 1360 |
| 新功能 | 1360 |
| 錯誤修復和其他更改 | 1360 |
| 文件歷史記錄 | 1362 |
| | mcccclxxi |

什麼是 Amazon DocumentDB (與 MongoDB 兼容性)

Amazon DocumentDB 資料庫 (與 MongoDB 兼容性) 是一種快速、可靠且全受管的資料庫服務。Amazon DocumentDB 可讓您輕鬆地在雲端中設定、操作和擴展與 MongoDB 相容的資料庫。透過 Amazon DocumentDB，您可以執行相同的應用程式程式碼，並使用與 MongoDB 搭配使用的相同驅動程式和工具。

在使用 Amazon DocumentDB 之前，您應該先檢閱中所述的概念和功能。[運作方式](#)接著再完成 [開始使用指南](#) 中的步驟。

主題

- [Amazon DocumentDB 概述](#)
- [叢集](#)
- [執行個體](#)
- [區域與可用區域](#)
- [Amazon DocumentDB 定價](#)
- [監控](#)
- [介面](#)
- [後續步驟？](#)
- [Amazon DocumentDB: 它是如何工作的](#)
- [什麼是文件資料庫？](#)

Amazon DocumentDB 概述

以下是 Amazon DocumentDB 的一些高級功能：

- Amazon DocumentDB 支援兩種類型的叢集：以執行個體為基礎的叢集和彈性叢集。彈性叢集支援工作負載，每秒可達數百萬次讀取/寫入，以及數 PB 的儲存容量。如需彈性叢集的詳細資訊，請參閱[使用 Amazon DocumentDB 彈性集群](#)。以下內容是指以 Amazon DocumentDB 執行個體為基礎的叢集。
- Amazon DocumentDB 會隨著您的資料庫儲存需求增加，自動增加儲存磁碟區的大小。您的儲存磁碟區以 10 GB 的增量增加，最高可達 128 TiB。您不需要為叢集佈建任何過多的儲存空間來處理未來的增長。

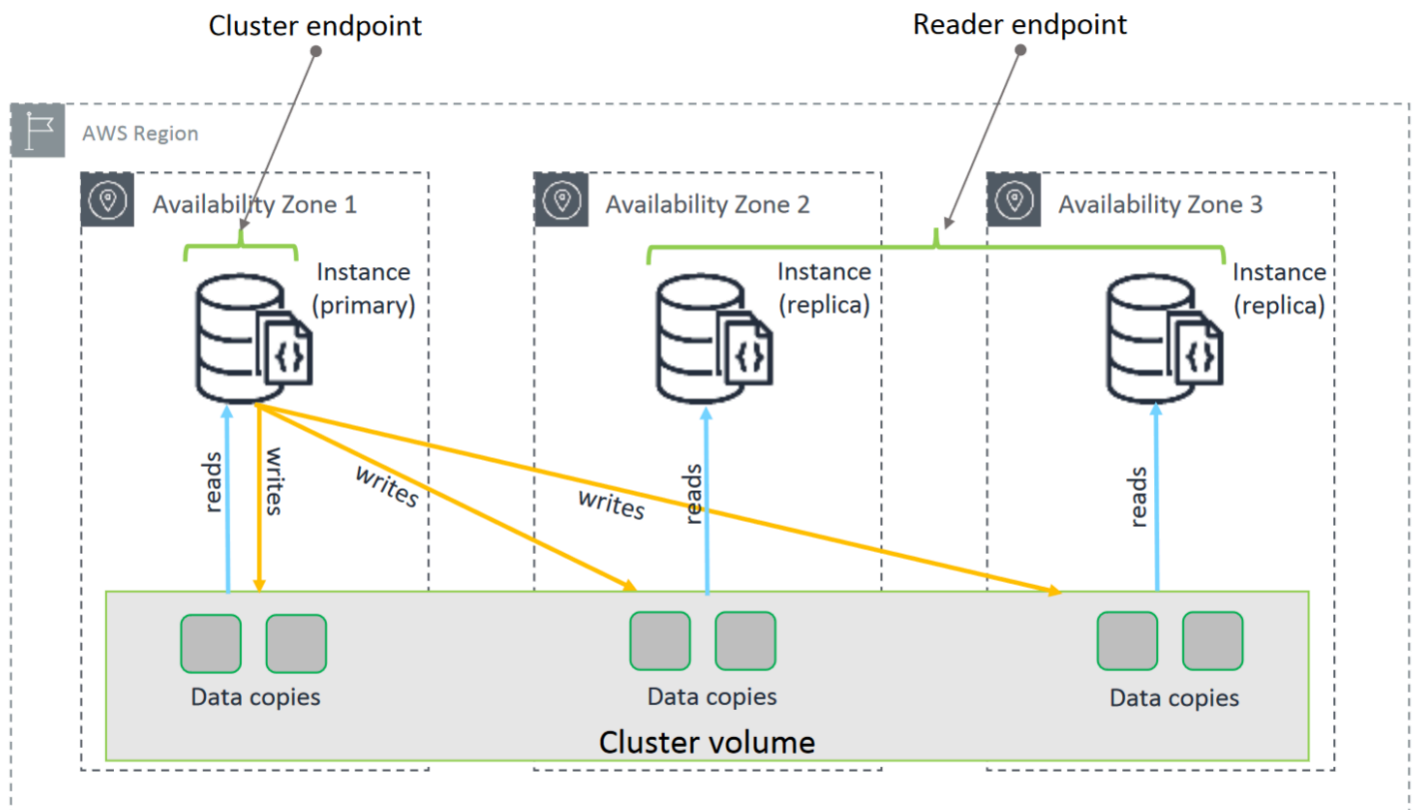
- 使用 Amazon DocumentDB，您可以建立最多 15 個複本執行個體，以增加讀取輸送量以支援大量應用程式請求。Amazon DocumentDB 複本共用相同的基礎儲存體，可降低成本並避免在複本節點執行寫入的需求。此功能可釋放更多的處理能力來處理讀取要求，並減少複本延遲時間，通常會縮短至毫秒數。無論儲存磁碟區大小為何，您都可以在幾分鐘內新增複本。Amazon DocumentDB 也提供讀取器端點，因此應用程式可以連線，而不必在新增和移除複本時追蹤複本。
- Amazon DocumentDB 可讓您擴展或縮減每個執行個體的運算和記憶體資源。運算擴展操作通常可在幾分鐘內完成。
- Amazon DocumentDB 在 Amazon Virtual Private Cloud (Amazon VPC) 中運行，因此您可以將數據庫隔離在自己的虛擬網絡中。您也可以設定防火牆設定，以控制您叢集的網路存取。
- Amazon DocumentDB 會持續監控叢集的運作狀態。在執行個體故障時，Amazon DocumentDB 會自動重新啟動執行個體和關聯的程序。Amazon DocumentDB 不需要資料庫重做日誌的當機復原重新執行，因此可大幅縮短重新啟動時間。Amazon DocumentDB 也會將資料庫快取與資料庫程序隔離，讓快取能夠在執行個體重新啟動時繼續存在。
- 在執行個體故障時，Amazon DocumentDB 會將容錯移轉自動化到您在其他可用區域中建立的多達 15 個 Amazon DocumentDB 複本之一。如果沒有佈建複本且發生故障，Amazon DocumentDB 會嘗試自動建立新的 Amazon DocumentDB 執行個體。
- Amazon DocumentDB 中的備份功能可讓您的叢集進行 point-in-time 復原。此功能可讓您將叢集還原至保留期間內任何一秒鐘的狀態，最多可恢復到前五分鐘的狀態。您可以設定最長 35 天的自動備份保留期。自動備份存放在 Amazon Simple Storage Service (Amazon S3) 中，其設計具有 99.999999999% 的耐久性。Amazon DocumentDB 備份是自動、增量和連續的備份，而且不會影響您的叢集效能。
- 使用 Amazon DocumentDB，您可以使用透過 AWS Key Management Service (AWS KMS) 建立和控制的金鑰來加密資料庫。在使用 Amazon DocumentDB 加密執行的資料庫叢集上，基礎儲存體中存放的靜態資料會加密。在相同叢集中的自動備份、快照及複本也會經過加密。

如果您不熟悉 AWS 服務，請使用下列資源進一步了解：

- AWS 提供運算、資料庫、儲存、分析及其他功能的服務。如需所有 AWS 服務的概觀，請參閱[使用 Amazon Web Services 進行雲端運算](#)。
- AWS 提供數個資料庫服務。如需有關哪種服務最適合您環境的指引，請參閱[上的資料庫 AWS](#)。

叢集

叢集包含 0 到 16 個執行個體，以及管理這些執行個體資料的叢集儲存磁碟區。所有寫入都是透過主要執行個體完成。所有執行個體 (主要和複本) 都支援讀取。叢集的資料儲存在叢集磁碟區中，其複本會存放在三個不同的可用區域。



Amazon DocumentDB 5.0 執行個體型叢集支援資料庫叢集的兩種儲存組態：Amazon DocumentDB 標準和 Amazon DocumentDB I/O 最佳化。如需更多資訊，請參閱[Amazon DocumentDB 群集存儲配置](#)。

執行個體

Amazon DocumentDB 執行個體是雲端中的隔離資料庫環境。執行個體可以包含多個使用者建立的資料庫。您可以使用 AWS Management Console 或建立和修改例證 AWS CLI。

執行個體的運算和記憶體容量是由其執行個體類別決定的。您可以選擇最符合您需求的執行個體。如果您的需求在一段時間後改變，您可以選擇其他執行個體類別。如需執行個體類別規格，請參閱[執行個體類別規格](#)。

Amazon DocumentDB 實例僅在 Amazon VPC 環境中運行。Amazon VPC 可讓您控制虛擬聯網環境：您可以選擇自己的 IP 地址範圍、建立子網路，以及設定路由和存取控制清單 (ACL)。

您必須先建立包含執行個體的叢集，才能建立 Amazon DocumentDB 執行個體。

並非每個區域都支援所有執行個體類別。下表顯示每個區域支援的執行個體類別。

各區域的支援執行個體類別

| 區域 | R6G | R5 | R4 | T4G | T3 |
|---------------|-----|----|----|-----|----|
| 美國東部 (俄亥俄) | 支援 | 支援 | 支援 | 支援 | 支援 |
| 美國東部 (維吉尼亞北部) | 支援 | 支援 | 支援 | 支援 | 支援 |
| 美國西部 (奧勒岡) | 支援 | 支援 | 支援 | 支援 | 支援 |
| 南美洲 (聖保羅) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (香港) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (海德拉巴) | | 支援 | | | 支援 |
| 亞太區域 (孟買) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (首爾) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (悉尼) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (新加坡) | 支援 | 支援 | | 支援 | 支援 |

| 區域 | R6G | R5 | R4 | T4G | T3 |
|---------------------|-----|----|----|-----|----|
| 亞太區域 (東京) | 支援 | 支援 | | 支援 | 支援 |
| 加拿大 (中部) | 支援 | 支援 | | 支援 | 支援 |
| 歐洲 (法蘭克福) | 支援 | 支援 | | 支援 | 支援 |
| 歐洲 (愛爾蘭) | 支援 | 支援 | 支援 | 支援 | 支援 |
| 歐洲 (倫敦) | 支援 | 支援 | | 支援 | 支援 |
| 歐洲 (米蘭) | 支援 | 支援 | | 支援 | 支援 |
| Europe (Paris) | 支援 | 支援 | | 支援 | 支援 |
| 中東 (阿拉伯聯合大公國) | 支援 | 支援 | | 支援 | 支援 |
| 中國 (北京) 區域 | 支援 | 支援 | | 支援 | 支援 |
| 中國 (寧夏) | 支援 | 支援 | | 支援 | 支援 |
| AWS GovCloud (美國西部) | 支援 | 支援 | | 支援 | 支援 |
| AWS GovCloud (美國東部) | 支援 | 支援 | | 支援 | 支援 |

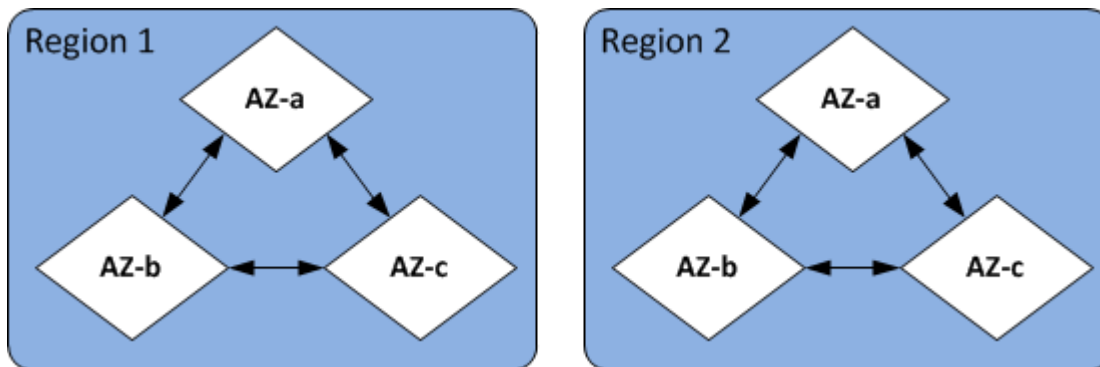
區域與可用區域

區域與可用區域定義叢集和執行個體的實體位置。

區域

AWS 雲計算資源存放在世界各地（例如北美，歐洲或亞洲）的高可用性數據中心設施中。各資料中心位置皆稱為區域。

每個 AWS 區域都被設計為與其他區 AWS 域完全隔離。每個區域中皆有多個可用區域。藉由在不同的可用區域中啟動您的節點，您可以實現最大的容錯能力。下圖顯示區域和可用區 AWS 域如何運作的高階檢視。



可用區域

每個區 AWS 域都包含稱為可用區域的多個不同位置。各個可用區域的設計，主要是為了與其他可用區域的故障事件隔離，並以低成本、低延遲的方式，透過網路連線至相同區域中的其他可用區域。藉由在多個可用區域中針對指定叢集啟動執行個體，您就可以保護應用程式免於發生可用區域故障的事件。

Amazon DocumentDB 架構將存儲和運算分開。對於儲存層，Amazon DocumentDB 會在三個 AWS 可用區域複寫六個資料副本。例如，如果您在僅支援兩個可用區域的區域中啟動 Amazon DocumentDB 叢集，則您的資料儲存將以六種方式複寫到三個可用區域，但是您的運算執行個體只能在兩個可用區域中使用。

下表列出可用於給定 AWS 區域 叢集佈建運算執行個體的可用區域數目。

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------|-----------|-----------|
| 美國東部 (俄亥俄) | us-east-2 | 3 |
| 美國東部 (維吉尼亞北部) | us-east-1 | 6 |
| 美國西部 (奧勒岡) | us-west-2 | 4 |

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------------|----------------|-----------|
| 南美洲 (聖保羅) | sa-east-1 | 3 |
| 亞太區域 (香港) | ap-east-1 | 3 |
| 亞太區域 (海德拉巴) | ap-south-2 | 3 |
| 亞太區域 (孟買) | ap-south-1 | 3 |
| 亞太區域 (首爾) | ap-northeast-2 | 4 |
| 亞太區域 (新加坡) | ap-southeast-1 | 3 |
| 亞太區域 (悉尼) | ap-southeast-2 | 3 |
| 亞太區域 (東京) | ap-northeast-1 | 3 |
| 加拿大 (中部) | ca-central-1 | 3 |
| 中國 (北京) 區域 | cn-north-1 | 3 |
| 中國 (寧夏) | cn-northwest-1 | 3 |
| 歐洲 (法蘭克福) | eu-central-1 | 3 |
| 歐洲 (愛爾蘭) | eu-west-1 | 3 |
| 歐洲 (倫敦) | eu-west-2 | 3 |
| 歐洲 (米蘭) | eu-south-1 | 3 |
| Europe (Paris) | eu-west-3 | 3 |
| 中東 (阿拉伯聯合大公國) | me-central-1 | 3 |
| AWS GovCloud (美國西部) | us-gov-west-1 | 3 |

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------------|---------------|-----------|
| AWS GovCloud (美國東部) | us-gov-east-1 | 3 |

Amazon DocumentDB 定價

Amazon DocumentDB 叢集是根據下列元件計費：

- 執行個體小時數 (每小時) — 以執行個體的執行個體類別為基礎 (例如, db.r5.xlarge)。定價以每小時為單位列出，但帳單已採用秒數為計算單位，並以十進位制顯示時間。Amazon DocumentDB 的使用量以一秒為增量計費，最少為 10 分鐘。如需詳細資訊，請參閱 [管理執行個體類別](#)。
- I/O 請求 (每月每 100 萬個請求) — 您在計費週期內發出的儲存 I/O 請求總數。
- B@@@ ackup 儲存體 (每月每 GiB) — Backup 儲存體是與自動資料庫備份以及您已建立的任何作用中資料庫快照集相關聯的儲存體。延長您的備份保留期或拍攝額外的資料庫快照，會增加資料庫所消耗的備份儲存。備份儲存是以 GB 月計算，因此每秒不適用。如需詳細資訊，請參閱 [在 Amazon DocumentDB 中備份和還原](#)。
- 資料傳輸 (每 GB) — 從網際網路或其他 AWS 區域傳入或傳出執行個體的資料。

如需詳細資訊，請參閱 [Amazon DocumentDB 定價](#)。

免費試用

您可以使用 1 個月的免費試用版免費試用 Amazon DocumentDB。如需詳細資訊，請參閱 [Amazon DocumentDB 定價中的免費試用版或參閱 Amazon DocumentDB 免費試用常見問答集](#)。

監控

追蹤執行個體的效能與運作狀態有多種方式。您可以使用免費的 Amazon CloudWatch 服務來監控執行個體的效能和運作狀態。您可以在 Amazon DocumentDB 主控台上找到效能圖表。您可以訂閱 Amazon DocumentDB 事件，以便在執行個體、快照、參數群組或安全群組發生變更時收到通知。

如需詳細資訊，請參閱下列內容：

- [使用監控亞馬遜文檔數據庫 CloudWatch](#)
- [使用記錄 Amazon DocumentDB API 呼叫AWS CloudTrail](#)

介面

有多種方式為您與 Amazon DocumentDB 進行交互，包括 AWS Management Console 和 AWS CLI

AWS Management Console

AWS Management Console 這是一個簡單的基於 Web 的用戶界面。不需編寫程式，即可從主控台管理您的叢集和執行個體。[要訪問 Amazon DocumentDB 控制台，請登錄 AWS Management Console 並打開 Amazon DocumentDB 控制台 `https://console.aws.amazon.com/docdb`。](#)

AWS CLI

您可以使用 AWS Command Line Interface (AWS CLI) 來管理您的 Amazon DocumentDB 叢集和執行個體。只要使用最少的組態，您就可以從最愛的終端機程式開始使用 Amazon DocumentDB 主控台提供的所有功能。

- 若要安裝 AWS CLI，請參閱[安裝指 AWS 命令行介面](#)。
- 要開始使用 Amazon DocumentDB，請參閱 Amazon DocumentDB 的[AWS 命令行界面參考](#)。AWS CLI

mongo 殼層

若要連線到叢集以建立、讀取、更新和刪除資料庫中的文件，您可以將mongo命令介面與 Amazon DocumentDB 搭配使用。若要下載並安裝 mongo 4.0 外殼，請參閱[第 4 步：安裝蒙戈外殼](#)。

MongoDB 驅動程式

若要針對 Amazon DocumentDB 叢集開發和寫入應用程式，您也可以使用 MongoDB 驅動程式搭配 Amazon DocumentDB。

後續步驟？

上述各節將向您介紹 Amazon DocumentDB 提供的基本基礎設施元件。接下來該怎麼做？視您的情況而定，請參閱下列其中一個主題以開始使用：

- 透過使用建立叢集和執行個體，開始使用 Amazon DocumentDB 料庫。[AWS CloudFormation Amazon DocumentDB 快速入門使用 AWS CloudFormation](#)
- 透過使用我們中的指示建立叢集和執行個體，開始使用 Amazon DocumentDB。[開始使用指南](#)

- 透過使用中的指示建立彈性叢集，開始使用 Amazon DocumentDB。 [開始使用 Amazon DocumentDB 彈性叢集](#)
- 使用以下指南將 MongoDB 實作遷移至 Amazon DocumentDB 料庫 [遷移到 Amazon DocumentDB](#)

Amazon DocumentDB: 它是如何工作的

Amazon DocumentDB 資料庫 (與 MongoDB 相容性) 是一個完全受管、與 MongoDB 相容的資料庫服務。透過 Amazon DocumentDB，您可以執行相同的應用程式程式碼，並使用與 MongoDB 搭配使用的相同驅動程式和工具。Amazon DocumentDB 與 MongoDB 3.6，4.0 和 5.0 兼容。

主題

- [Amazon DocumentDB 端點](#)
- [TLS 支援](#)
- [Amazon DocumentDB 存儲](#)
- [Amazon DocumentDB 複寫](#)
- [Amazon DocumentDB 可靠性](#)
- [讀取偏好設定選項](#)
- [TTL 刪除](#)
- [可計費資源](#)

當您使用 Amazon DocumentDB 時，首先要創建一個叢集。資料庫叢集包含零或多個資料庫執行個體，和一個管理這些執行個體資料的叢集磁碟區。Amazon DocumentDB 叢集磁碟區是跨越多個可用區域的虛擬資料庫儲存磁碟區。每個可用區域具有叢集資料的複本。

Amazon DocumentDB 集群由兩個組件組成：

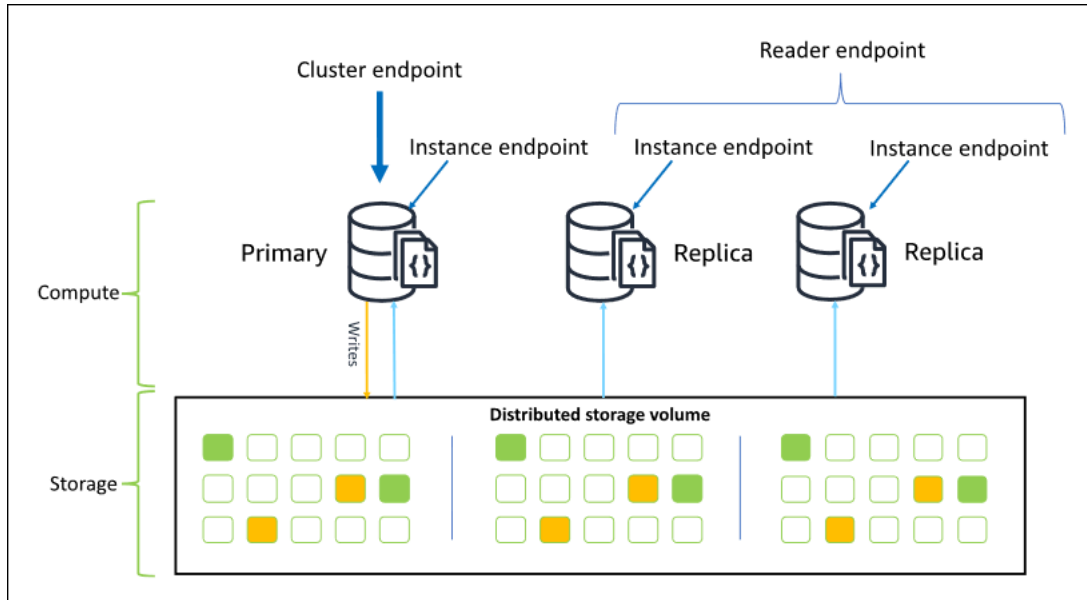
- 叢集磁碟區：使用雲端原生儲存服務，跨三個可用區域六種方式複寫資料，提供高耐用性和可用性的儲存體。Amazon DocumentDB 叢集只有一個叢集磁碟區，最多可存放 128 TiB 的資料。
- 執行個體 — 提供資料庫的處理能力，將資料寫入叢集儲存磁碟區，以及從叢集儲存磁碟區讀取資料。一個 Amazon DocumentDB 集群可以有 0-16 個實例。

執行個體提供以下兩種角色：

- 主要執行個體：支援讀取和寫入作業，並對叢集磁碟區執行所有資料修改。每個 Amazon DocumentDB 叢集都有一個主要執行個體。

- 複本執行個體 — 僅支援讀取作業。除了主執行個體之外，Amazon DocumentDB 叢集最多可以有 15 個複本。擁有多個複本可讓您分配讀取工作負載。此外，透過將複本置於不同可用區域，您也可提高叢集可用性。

下圖說明 Amazon DocumentDB 叢集中叢集磁碟區、主要執行個體和複本之間的關係：



叢集執行個體不需為相同執行個體類別，而您可以隨需要佈建和終止它們。這個架構可讓您擴展叢集的運算容量，而不影響儲存。

當您的應用程式會將資料寫入主執行個體，主執行個體會將耐用資料寫入叢集磁碟區。然後，它會將該寫入狀態 (而非資料) 複寫到每個使用中複本。Amazon DocumentDB 複本不會參與處理寫入，因此 Amazon DocumentDB 複本對於讀取擴展有利。從 Amazon DocumentDB 複本讀取最終會與最小的複本延遲一致，通常在主執行個體寫入資料之後不到 100 毫秒。從複本的讀取保證會以複本寫入主要執行個體的順序來進行讀取。複本延遲取決於資料變更的速率，以及高寫入活動的期間可能會增加複本延遲。如需詳細資訊，請參閱 [Amazon DocumentDB 指標](#) 的 ReplicationLag 指標。

Amazon DocumentDB 端點

Amazon DocumentDB 提供多種連線選項，可為各種使用案例提供服務。若要連接到 Amazon DocumentDB 叢集中的執行個體，請指定執行個體的端點。端點是主機地址和連接埠號碼 (以冒號分隔)。

除非您有連線至讀取器端點或執行個體端點的特定使用案例，否則我們建議使用叢集端點並在複本集模式下 (請參閱 [以複本集形式連接至 Amazon DocumentDB](#)) 連線至叢集。若要將請求路由到您的

複本，請選擇驅動程式讀取偏好設定，以發揮最大讀取擴展，同時符合您應用程式的讀取一致性要求。secondaryPreferred 讀取偏好設定會啟用複本讀取，並釋出主要執行個體以進行更多工作。

下列端點可從 Amazon DocumentDB 叢集取得。

叢集端點

叢集端點會連接至叢集目前的主要執行個體。您可以使用叢集端點來執行讀取和寫入操作。亞 Amazon DocumentDB 叢集只有一個叢集端點。

叢集端點可為叢集的讀寫連接提供容錯移轉支援。如果叢集目前的主要執行個體發生故障，您的叢集至少有一個作用中的僅供讀取複本，叢集端點會自動重新導向到新主要執行個體的連線請求。連線到 Amazon DocumentDB 叢集時，建議您使用叢集端點並以複本集模式連線到叢集 (請參閱[以複本集形式連接至 Amazon DocumentDB](#))。

以下是 Amazon DocumentDB 叢集端點的示例：

```
sample-cluster.cluster-123456789012.us-east-1.docdb.amazonaws.com:27017
```

以下是使用此叢集端點的範例連線字串：

```
mongodb://username:password@sample-cluster.cluster-123456789012.us-east-1.docdb.amazonaws.com:27017
```

如需尋找叢集端點的詳細資訊，請參閱 [尋找叢集端點](#)。

讀取器端點

讀取器端點負載會平衡在叢集中所有可用複本的僅供讀取連線。嘗試透過連接至讀取器端點來執行寫入操作發生錯誤。Amazon DocumentDB 叢集只有一個讀取器端點。

如果叢集僅包含一個 (主要) 執行個體，讀取器端點會連接至主要執行個體。將複本執行個體新增至 Amazon DocumentDB 叢集時，讀取器端點會在新複本處於作用中狀態後開啟與該複本的唯一讀連線。

以下是 Amazon DocumentDB 叢集的讀取器端點範例：

```
sample-cluster.cluster-ro-123456789012.us-east-1.docdb.amazonaws.com:27017
```

以下是使用此讀取器端點的範例連線字串：

```
mongodb://username:password@sample-cluster.cluster-ro-123456789012.us-east-1.docdb.amazonaws.com:27017
```

讀取器端點負載會平衡唯讀連線，而不是讀取請求。如果一些讀取器端點連線的使用量較其他連線頻繁，則系統可能無法在叢集中的執行個體間平均地平衡讀取請求。建議連線至做為複本集的叢集端點，並利用 `secondaryPreferred` 讀取偏好設定選項，來發佈請求。

如需尋找叢集端點的詳細資訊，請參閱 [尋找叢集端點](#)。

執行個體端點

執行個體端點會連線到叢集中的特定執行個體。您可以將目前主要執行個體的執行個體端點用於讀取和寫入操作。不過，嘗試對執行個體端點執行讀取複本的寫入操作可能會造成錯誤。Amazon DocumentDB 叢集的每個作用中執行個體都有一個執行個體端點。

執行個體端點對特定執行個體的連接提供直接控制，使用叢集端點或讀取器端點的案例可能不適用。範例使用案例佈建的是適用於週期性唯讀分析工作負載。您可以佈建 `larger-than-normal` 複本執行個體、使用其執行個體端點直接連線至新的較大型執行個體、執行分析查詢，然後終止執行個體。使用執行個體端點可讓分析流量不會影響到其他叢集執行個體。

以下是 Amazon DocumentDB 叢集中單一執行個體的執行個體端點範例：

```
sample-instance.123456789012.us-east-1.docdb.amazonaws.com:27017
```

以下是使用此執行個體端點的範例連線字串：

```
mongodb://username:password@sample-instance.123456789012.us-east-1.docdb.amazonaws.com:27017
```

Note

做為主要或複本的執行個體角色可以因為容錯移轉事件而變更。應用程式絕不應該假設特定執行個體端點是主執行個體。我們不建議連線至生產應用程式的執行個體端點。我們建議您改為使用叢集端點並在複本集模式下 (請參閱 [以複本集形式連接至 Amazon DocumentDB](#)) 連接至您的叢集。如需執行個體容錯移轉優先順序更進階的控制，請參閱 [了解 Amazon DocumentDB 叢集容錯](#)。

如需尋找叢集端點的詳細資訊，請參閱 [搜尋執行個體的端點](#)。

複本集模式

您可以指定複本集名稱，以複本集模式連接到 Amazon DocumentDB 叢集端點。rs0 在複本集模式中連接可讓您指定 Read Concern、Write Concern 和 Read Preference 選項。如需詳細資訊，請參閱 [讀取一致性](#)。

以下是在複本集模式中連接的範例連線字串：

```
mongodb://username:password@sample-cluster.cluster-123456789012.us-east-1.docdb.amazonaws.com:27017/?replicaSet=rs0
```

當您以複本集模式連線時，您的 Amazon DocumentDB 叢集會以複本集的形式顯示給驅動程式和用戶端。從 Amazon DocumentDB 叢集新增和移除的執行個體會自動反映在複本集組態中。

每個 Amazon DocumentDB 叢集都包含具有預設名稱的單一複本集。rs0 複本集名稱無法修改。

若是一般用途，建議連接到在複本集模式中的叢集端點。

Note

Amazon DocumentDB 叢集中的所有執行個體都會在相同的 TCP 連接埠上接聽以進行連線。

TLS 支援

如需使用傳輸層安全性 (TLS) 連線至 Amazon DocumentDB 的詳細資訊，請參閱 [對傳輸中資料進行加密](#)

Amazon DocumentDB 存儲

Amazon DocumentDB 資料存放在叢集磁碟區中，叢集磁碟區是使用固態硬碟 (SSD) 的單一虛擬磁碟區。叢集磁碟區由六個資料副本組成，這些副本會在單一的多個可用區域間自動複寫 AWS 區域。此複寫有助於確保您的資料具有高耐用性，同時降低資料遺失的機率。它也有助於確保您的叢集在容錯移轉時提供更高的可用性，因為資料副本已存在於其他可用區域。這些副本可以繼續為 Amazon DocumentDB 叢集中的執行個體提供資料請求。

資料儲存體的計費方式

隨著資料量的增加，Amazon DocumentDB 會自動增加叢集磁碟區的大小。Amazon DocumentDB 叢集磁碟區的大小上限為 128 TiB；不過，您只需支付在 Amazon DocumentDB 叢集磁碟區中使用的空

間費用。從 Amazon DocumentDB 4.0 開始，當資料被移除 (例如透過刪除集合或索引) 時，整體配置的空間會減少可比較的數量。因此，您可以刪除不再需要的集合、索引和資料庫，以減少儲存體費用。使用 Amazon DocumentDB 3.6 時，移除資料時 (例如透過刪除集合或索引)，整體配置的空間會保持不變。future 資料量增加時，系統會自動重複使用可用空間。

Note

使用 Amazon DocumentDB 3.6 時，儲存成本是根據儲存「高水位標記」(在任何時間點為 Amazon DocumentDB 叢集配置的最大量)。您可以避免建立大量臨時資訊的 ETL 做法，或在移除不需要的舊資料之前載入大量新資料，藉此管理成本。如果從 Amazon DocumentDB 叢集移除資料會導致大量已配置但未使用的空間，則重設高浮水印需要執行邏輯資料傾印，並使用或等工具還原到新叢集。mongodump mongorestore 建立和還原快照不會減少配置的儲存空間，因為基礎儲存的實體配置在還原的快照中保持不變。

Note

使用 mongodump 和 mongorestore 之類的公用程式會根據所讀取和寫入儲存磁碟區的資料大小產生 I/O 費用。

[如需 Amazon DocumentDB 資料儲存和 I/O 定價的相關資訊，請參閱 Amazon 文件資料庫 \(與 MongoDB 相容性\) 定價和定價常見問答集。](#)

Amazon DocumentDB 複寫

在 Amazon DocumentDB 叢集中，每個複本執行個體都會公開一個獨立的端點。這些複本端點對叢集磁碟區的資料提供唯讀存取權。它們可讓您隨著多個複寫執行個體擴展資料的讀取工作負載。它們還有助於提高資料讀取的效能，並提高 Amazon DocumentDB 叢集中資料的可用性。Amazon DocumentDB 複本也是容錯移轉目標，如果 Amazon DocumentDB 叢集的主執行個體發生故障，就會快速升級。

Amazon DocumentDB 可靠性

Amazon DocumentDB 的設計是為了可靠、耐用和容錯。若要提高可用性，您應該設定 Amazon DocumentDB 叢集，以便在不同的可用區域中擁有多個複本執行個體。) Amazon DocumentDB 包含數種自動功能，使其成為可靠的資料庫解決方案。

儲存體自動修復

Amazon DocumentDB 會在三個可用區域中維護多個資料副本，大幅減少因儲存故障而遺失資料的機會。Amazon DocumentDB 會自動偵測叢集磁碟區中的故障。當叢集磁碟區的某個區段發生故障時，Amazon DocumentDB 會立即修復該區段。它會使用叢集磁碟區組成之其他磁碟區中的資料，以協助確保中修復區段的資料是最新的。因此，Amazon DocumentDB 可避免資料遺失，並減少執行還原以從執行個體故障中 point-in-time 復原的需求。

可存活的快取預備

Amazon DocumentDB 會在與資料庫不同的程序中管理其頁面快取，因此頁面快取可以獨立存在於資料庫之外。在極少的情況下，若資料庫發生故障，頁面快取仍會留在記憶體中。這可確保在資料庫重新啟動時緩衝集區仍是最新狀態。

損毀復原

Amazon DocumentDB 的設計目的是幾乎可以立即從損毀中復原，並繼續為您的應用程式資料提供服務。Amazon DocumentDB 會在 parallel 執行緒上以非同步方式執行損毀復原，因此您的資料庫在當機後幾乎可以立即開啟且可用。

資源治理

Amazon DocumentDB 可保護在服務中執行關鍵流程所需的資源，例如運作狀態檢查。為此，當執行個體遇到高記憶體壓力時，Amazon DocumentDB 會限制請求。因此，某些作業可能會排入佇列，等待記憶體壓力消退。如果記憶體壓力持續，佇列的作業可能會逾時。您可以使用下列 CloudWatch 指標來監視是否因記憶體不足而導致的服務節流作業：LowMemThrottleQueueDepth、LowMemThrottleMaxQueueDepth、LowMemNumOperationsThrottled。如需詳細資訊，請參閱使用 CloudWatch。如果您看到執行個體持續存在的記憶體壓力，我們建議您擴大執行個體，為您的工作負載提供額外的記憶體。LowMem CloudWatch

讀取偏好設定選項

Amazon DocumentDB 使用雲端原生共用儲存服務，在三個可用區域複寫資料六次，以提供高耐用性。Amazon DocumentDB 不需要將資料複寫到多個執行個體來達到耐久性。無論您的叢集資料包含的是單一執行個體或 15 個執行個體，它都具有耐用性。

寫入耐久性

Amazon DocumentDB 使用獨特、分散式、容錯、自我修復的儲存系統。此系統會在三個 AWS 可用區域複製六份資料副本 (V=6)，以提供高可用性和耐久性。寫入資料時，Amazon DocumentDB 可確保在

確認寫入用戶端之前，在大多數節點上持久記錄所有寫入作業。如果您正在執行三節點 MongoDB 複本集，與 Amazon DocumentDB 進行比較時，使用的寫入問題 `{w:3, j:true}` 會產生最佳的組態。

對 Amazon DocumentDB 叢集的寫入必須由叢集的寫入執行個體處理。嘗試寫入讀取器會導致錯誤。來自 Amazon DocumentDB 主要執行個體的可寫入資訊是耐用的，而且無法復原。Amazon DocumentDB 預設為高耐用性，不支援非耐用寫入選項。您無法修改耐用性層級 (即寫入考量)。Amazon DocumentDB 忽略 `w = 任何東西`，並且是有效的 `w : 3` 和 `j : 真`。你不能減少它。

由於 Amazon DocumentDB 架構中的儲存和運算是分開的，因此具有單一執行個體的叢集非常耐用。耐用性會在儲存層處理。因此，具有單一執行個體和一個具有三個執行個體的 Amazon DocumentDB 叢集可達到相同等級的持久性。您可以設定您的叢集，以適用於特定使用案例，同時又可為您的資料提供高耐用性。

對 Amazon DocumentDB 叢集的寫入在單一文件中是不可原子的。

Amazon DocumentDB 不支援此選 `wtimeout` 項，如果指定了值，也不會傳回錯誤。保證不會無限期封鎖對主要 Amazon DocumentDB 執行個體的寫入。

讀取隔離

從 Amazon DocumentDB 執行個體讀取，只會傳回在查詢開始之前持久耐用的資料。讀取永遠不會傳回在查詢開始執行後修改的資料，在任何情況下。也不可能發生已變更讀取。

讀取一致性

從 Amazon 文件資料庫叢集讀取的資料非常耐用，而且不會復原。您可以透過指定請求或連線的讀取偏好設定來修改 Amazon DocumentDB 讀取的讀取一致性。Amazon DocumentDB 不支持非持久性讀取選項。

在正常操作條件下，從 Amazon DocumentDB 叢集的主要執行個體讀取會保持一致性，且具有 `read-after-write` 一致性。如果在寫入和後續讀取之間發生容錯移轉事件，系統可以短暫傳回非極其一致的讀取。所有來自僅供讀取複本的讀取最終皆會一致，並以相同順序傳回資料，複本延遲通常小於 100 毫秒。

Amazon DocumentDB 讀取首選項

只有在以複本集模式從叢集端點讀取資料時，Amazon DocumentDB 才支援設定讀取偏好設定選項。設定讀取偏好設定選項會影響 MongoDB 用戶端或驅動程式將讀取請求路由到 Amazon DocumentDB 叢集中執行個體的方式。您可以為特定查詢設定讀取偏好設定選項，或將其設為 MongoDB 驅動程式的一般選項。(請參閱用戶端或驅動程式的文件，以取得如何設定讀取偏好設定選項。)

如果您的用戶端或驅動程式未以複本集模式連線至 Amazon DocumentDB 叢集端點，則指定讀取偏好設定的結果未定義。

Amazon DocumentDB 不支援將標籤組設定為讀取偏好設定。

支援讀取偏好設定選項

- **primary** 指定 primary 讀取偏好設定有助於確保所有讀取都路由至叢集的主要執行個體。如果主要執行個體不可用時，讀取操作會失敗。primary 讀取偏好設定可產生 read-after-write 一致性，並適用於將 read-after-write 一致性優先於高可用性和讀取擴展的使用案例。

以下範例指定 primary 讀取偏好設定：

```
db.example.find().readPref('primary')
```

- **primaryPreferred** 指定 primaryPreferred 讀取偏好設定路由會在正常操作下讀取主要實例。如果是主要容錯移轉，用戶端會將請求路由到複本。primaryPreferred 讀取偏好設定可在正常作業期間產生一致 read-after-write 性，並在容錯移轉事件期間最終一致讀取。primaryPreferred 讀取偏好設定適用於將 read-after-write 一致性優先於讀取擴展，但仍需要高可用性的使用案例。

以下範例指定 primaryPreferred 讀取偏好設定：

```
db.example.find().readPref('primaryPreferred')
```

- **secondary** 指定 secondary 讀取偏好設定可確保只會將讀取路由至複本，而不會傳送至主要實例。如果叢集中沒有複本執行個體，讀取請求會失敗。secondary 讀取偏好設定會產生最終一致的讀取，適用於將主要執行個體寫入輸送量優先於高可用 read-after-write 性和一致性的使用案例。

以下範例指定 secondary 讀取偏好設定：

```
db.example.find().readPref('secondary')
```

- **secondaryPreferred** 指定 secondaryPreferred 讀取偏好設定可確保當一或多個複本處於使用中狀態時，讀取會路由至僅供讀取複本。如果沒有叢集中沒有作用中的複本執行個體，系統會將讀取請求路由到主要執行個體。secondaryPreferred 讀取偏好會在僅供讀取複本提供讀取服務時，

提供最終一致讀取。當主要執行個體提供讀取服務時，它會產生 read-after-write 一致性 (禁止容錯移轉事件)。secondaryPreferred 讀取偏好設定適用於將讀取擴展與高可用性優先於 read-after-write 一致性的使用案例。

以下範例指定 secondaryPreferred 讀取偏好設定：

```
db.example.find().readPref('secondaryPreferred')
```

- **nearest**— 指定 nearest 讀取偏好設定僅根據用戶端與 Amazon DocumentDB 叢集中所有執行個體之間測量的延遲來路由讀取。nearest 讀取偏好會在僅供讀取複本提供讀取服務時，提供最終一致讀取。當主要執行個體提供讀取服務時，它會產生 read-after-write 一致性 (禁止容錯移轉事件)。nearest 讀取偏好設定適用於優先順序達到最低可能讀取延遲和高可用性，而不是 read-after-write 一致性和讀取擴展的使用案例。

以下範例指定 nearest 讀取偏好設定：

```
db.example.find().readPref('nearest')
```

高可用性

Amazon DocumentDB 透過使用複本做為主要執行個體的容錯移轉目標來支援高可用性叢集組態。如果主執行個體失敗，Amazon DocumentDB 複本會提升為新的主執行個體，並會短暫中斷，在此期間，對主執行個體所做的讀取和寫入請求會失敗，但發生例外狀況。

如果您的 Amazon DocumentDB 叢集不包含任何複本，則會在發生故障時重新建立主執行個體。不過，升級 Amazon DocumentDB 複本比重新建立主執行個體快得多。因此，我們建議您建立一或多個 Amazon DocumentDB 複本做為容錯移轉目標。

做為容錯移轉目標的複本執行個體類別應與主要執行個體的執行個體類別相同。它們應該佈建在與主要執行個體不同的可用區域。您可以控制哪些複本是偏好做為容錯移轉目標。如需設定 Amazon DocumentDB 以取得高可用性的最佳實務，請參閱 [了解 Amazon DocumentDB 叢集容錯](#)

擴展讀取

Amazon DocumentDB 複本是讀取擴展的理想選擇。他們完全專用於叢集磁碟區上的讀取操作，也就是說，複本並不會處理寫入。資料複寫發生於叢集磁碟區內，而非執行個體之間。因此，每個複本的資源都專用於處理您的查詢，而非複寫和寫入資料。

如果您的應用程式需要讀取容量，您可以快速將複本新增到您的叢集 (通常少於十分鐘)。如果您的讀取容量需求減少，您可以移除不需要的複本。使用 Amazon DocumentDB 複本時，您只需為所需的讀取容量付費。

Amazon DocumentDB 透過使用讀取偏好設定選項來支援用戶端讀取擴展。如需詳細資訊，請參閱 [Amazon DocumentDB 讀取首選項](#)。

TTL 刪除

透過背景程序從 TTL 索引刪除是最佳作法，但無法保證可在特定時間範圍內完成。像是執行個體大小、執行個體資源使用率、文件大小和整體輸送量等因素，都會影響 TTL 刪除的時間。

當 TTL 監控器刪除文件時，每個刪除都會產生 IO 成本，使帳單金額增加。如果輸送量和 TTL 刪除率增加，請注意帳單金額會因為 IO 使用率提升而增加。

當您在現有集合上建立 TTL 索引時，必須先刪除所有過期的文件，然後再建立索引。目前的 TTL 實作已針對刪除集合中的一小部分文件進行最佳化，如果從一開始就在集合上啟用 TTL，這很典型，如果需要一次性刪除大量文件，則可能會導致高於必要的 IOPS。

如果您不想建立 TTL 索引來刪除文件，您可以改為根據時間將文件細分為集合，並在不再需要文件時直接拖放這些集合。例如：您可以每週建立一個集合，然後將其刪除，而不會產生 IO 成本。這可能比使用 TTL 索引更具成本效益。

可計費資源

識別可計費的 Amazon DocumentDB 資源

Amazon DocumentDB 是一項完全受管的資料庫服務，會針對執行個體、儲存、I/O、備份和資料傳輸收費。如需詳細資訊，請參閱 [Amazon DocumentDB \(與 MongoDB 相容性\) 定價](#)。

若要探索帳戶中的可計費資源，並可能刪除資源，您可以使用 AWS Management Console 或 AWS CLI。

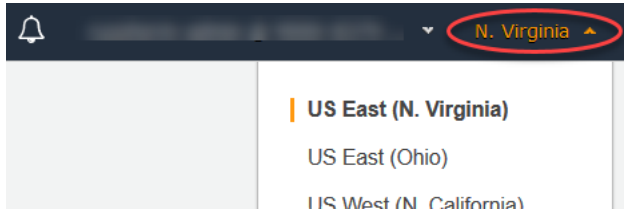
使用 AWS Management Console

使用 AWS Management Console，您可以探索已為指定佈建的 Amazon DocumentDB 叢集、執行個體和快照。AWS 區域

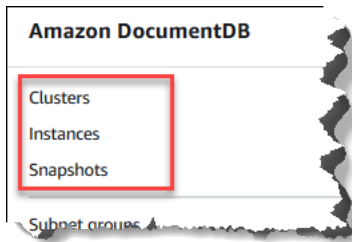
找出叢集、執行個體和快照

1. [登入 AWS Management Console](#)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。

- 若要探索預設地區以外的地區的可計費資源，請在畫面右上角選擇您 AWS 區域 要搜尋的資源。



- 在導覽窗格中，選擇您感興趣的應計費資源類型：Clusters (叢集)、Instances (執行個體) 或 Snapshots (快照)。



- 右窗格會列出您針對該區域佈建的所有叢集、執行個體或快照。您需要支付叢集、執行個體和快照的費用。

使用 AWS CLI

使用 AWS CLI，您可以探索已為指定佈建的 Amazon DocumentDB 叢集、執行個體和快照。AWS 區域

找出叢集和執行個體

以下程式碼會列出指定區域的所有叢集和執行個體。如果想要搜尋預設區域中的叢集和執行個體，您可以省略 `--region` 參數。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \
  --region us-east-1 \
  --query 'DBClusters[?Engine==`docdb`]' | \
  grep -e "DBClusterIdentifier" -e "DBInstanceIdentifier"
```

針對 Windows：

```
aws docdb describe-db-clusters ^
```

```
--region us-east-1 ^
--query 'DBClusters[?Engine==`docdb`] | ^
grep -e "DBClusterIdentifier" -e "DBInstanceIdentifier"
```

此操作的輸出將會如下所示。

```
"DBClusterIdentifier": "docdb-2019-01-09-23-55-38",
  "DBInstanceIdentifier": "docdb-2019-01-09-23-55-38",
  "DBInstanceIdentifier": "docdb-2019-01-09-23-55-382",
"DBClusterIdentifier": "sample-cluster",
"DBClusterIdentifier": "sample-cluster2",
```

找出快照

以下程式碼會列出指定區域的所有快照。如果想要搜尋預設區域中的快照，您可以省略 `--region` 參數。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-snapshots \
--region us-east-1 \
--query 'DBClusterSnapshots[?Engine==`docdb`].
[DBClusterSnapshotIdentifier,SnapshotType]'
```

針對 Windows：

```
aws docdb describe-db-cluster-snapshots ^
--region us-east-1 ^
--query 'DBClusterSnapshots[?Engine==`docdb`].
[DBClusterSnapshotIdentifier,SnapshotType]'
```

此操作的輸出將會如下所示。

```
[
  [
    "rds:docdb-2019-01-09-23-55-38-2019-02-13-00-06",
    "automated"
  ],
  [
    "test-snap",
    "manual"
  ]
]
```



```
]
]
```

您只需要刪除 manual 快照。當您刪除叢集時，也會刪除 Automated 快照。

刪除不需要的應計費資源

若要刪除某個叢集，您必須先刪除該叢集中的所有執行個體。

- 若要刪除執行個體，請參閱[刪除 Amazon DocumentDB 實例](#)。

Important

即使您刪除叢集中的執行個體，您仍需要支付使用該叢集相關聯儲存和備份的費用。若要停止所有費用，您必須也要刪除您的叢集和手動快照。

- 若要刪除叢集，請參閱[刪除 Amazon DocumentDB 集群](#)。
- 若要刪除手動快照，請參閱[刪除叢集快照](#)。

什麼是文件資料庫？

有些開發人員不是以正規化的列和欄來思考其資料模型。通常在應用程式層級，資料是以 JSON 文件來表示，因為開發人員以文件形式來思考資料模式更為直覺。

文件資料庫的普遍性一直在成長，因為它們可讓您使用和您的應用程式碼相同的文件模型格式，將資料存留於資料庫中。文件資料庫提供功能強大且直覺式的 API 進行靈活彈性開發。

主題

- [文件資料庫使用案例](#)
- [了解文件](#)
- [使用文件](#)

文件資料庫使用案例

您的使用案例會影響您需要的是文件資料庫，或是其他類型的資料庫來管理您的資料。文件資料庫對於需要靈活結構描述以便快速進行重複性開發的工作負載來說非常適合。以下是一些使用案例的範例，文件資料庫可為這些使用案例提供相當明顯的優勢：

主題

- [使用者設定檔](#)
- [即時大數據](#)
- [內容管理](#)

使用者設定檔

由於文件資料庫擁有靈活的結構描述，因此能夠存放擁有不同屬性和資料值的文件。文件資料庫對於不同使用者提供不同類型資訊的線上設定檔來說，是相當實用的解決方案。您可以使用文件資料庫單獨存放每個使用者特有的屬性，藉此有效率地存放每個使用者的設定檔。

假設使用者選擇新增或移除其設定檔中的資訊。在這種情況下，就可輕鬆將其文件取代為更新版本，該版本當中會包含任何最近新增的屬性和資料，或省略任何近期省略的屬性和資料。文件資料庫可輕鬆管理此層級的個別特性和流暢度。

即時大數據

在過去，由於營運資料庫和分析資料庫分別在不同環境中維護的事實，而使得從營運資料當中蒐集資訊的能力受到阻礙，也就是說，營運和業務/報告分別進行。在競爭激烈的商業環境中，即時擷取營運資訊的能力至關重要。透過使用文件資料庫，企業就能存放和管理任何來源的營運資料，並同時將資料饋送至選擇的 BI 引擎進行分析。但並不需要有兩個環境。

內容管理

為了有效管理內容，您必須能夠收集和彙總各種不同來源的內容，然後將內容傳遞給客戶。文件資料庫採用靈活的結構描述，是收集和存放任何類型資料最理想的選擇。您可以使用文件資料庫來建立和納入新的內容類型，包括使用者產生的內容，例如映像、評論和影片。

了解文件

文件資料庫用於將半結構化的資料儲存為文件而不是跨多個各自具有獨特且固定結構的表格將資料正規化，如同關聯式資料庫一樣。存放在文件資料庫中的文件會使用巢狀索引鍵值組提供文件的結構或結構描述。不過，不同類型的文件可以存放在相同的文件資料庫中，因此符合處理不同格式的類似資料時的需求。例如，由於每個文件是自我描述的，因此[文件資料庫中文件的範例](#)主題所述線上商店的 JSON 編碼文件可以存放在相同的文件資料庫中。

主題

- [SQL 與非關聯式術語](#)

- [簡易文件](#)
- [內嵌文件](#)
- [文件資料庫中文件的範例](#)
- [了解文件資料庫正規化](#)

SQL 與非關聯式術語

下表比較文件資料庫 (MongoDB) 所用的術語與 SQL 資料庫所用的術語。

| SQL | MongoDB |
|---------|----------|
| 資料表 | 收集 |
| 列 | 文件 |
| 資料欄 | 欄位 |
| 主索引鍵 | ObjectId |
| 索引 | 索引 |
| 檢視 | 檢視 |
| 巢狀表格或物件 | 內嵌文件 |
| Array | Array |

簡易文件

文件資料庫中的所有文件都是自我描述。本文採用類似 JSON 格式的文件，但您可以使用其他編碼方式。

簡易文件有一或多個欄位，這些欄位在文件中全都位於同一層級。在下列範例中，欄位 SSN、LName、FName、DOB、Street、City、State-Province、PostalCode 和 Country 在文件中全都是同級。

```
{
  "SSN": "123-45-6789",
```

```
"LName": "Rivera",
"FName": "Martha",
"DOB": "1992-11-16",
"Street": "125 Main St.",
"City": "Anytown",
"State-Province": "WA",
"PostalCode": "98117",
"Country": "USA"
}
```

將資訊整理到簡易文件中時，每個欄位都會分別管理。若要擷取人員的地址，您必須擷取 Street、City、State-Province、PostalCode 和 Country 做為個別資料項目。

內嵌文件

複雜文件會透過在文件中建立內嵌文件的方式組織其資料。內嵌文件有助於管理分組資料和個別資料項目，端看有所指案例中哪一種方式較有效率。使用上述範例可在主文件中內嵌 Address 文件。這樣做會產生以下文件結構：

```
{
  "SSN": "123-45-6789",
  "LName": "Rivera",
  "FName": "Martha",
  "DOB": "1992-11-16",
  "Address":
  {
    "Street": "125 Main St.",
    "City": "Anytown",
    "State-Province": "WA",
    "PostalCode": "98117",
    "Country": "USA"
  }
}
```

您現在可以單獨的欄位存取文件中的資料 ("SSN":)，作為嵌入式文檔 ("Address":)，或作為嵌入式文檔的成員 ("Address":{"Street":})。

文件資料庫中文件的範例

如前面所述，由於文件資料庫中每個文件都是自我描述，因此文件資料庫內文件的結構可能各有不同。以下兩個文件，一個用於書籍，另一個用於期刊，兩者結構有所不同。然而兩個文件可以存放在相同文件資料庫中。

以下是範例書籍文件：

```
{
  "_id" : "9876543210123",
  "Type": "book",
  "ISBN": "987-6-543-21012-3",
  "Author":
  {
    "LName": "Roe",
    "MI": "T",
    "FName": "Richard"
  },
  "Title": "Understanding Document Databases"
}
```

以下是有兩篇文章的範例期刊文件：

```
{
  "_id" : "0123456789012",
  "Publication": "Programming Today",
  "Issue":
  {
    "Volume": "14",
    "Number": "09"
  },
  "Articles" : [
    {
      "Title": "Is a Document Database Your Best Solution?",
      "Author":
      {
        "LName": "Major",
        "FName": "Mary"
      }
    },
    {
      "Title": "Databases for Online Solutions",
      "Author":
      {
        "LName": "Stiles",
        "FName": "John"
      }
    }
  ],
}
```

```
"Type": "periodical"  
}
```

比較這兩個文件的結構。使用關聯式資料庫時，您需要使用個別的「期刊」和「書籍」表格，或是單一表格，其中未使用的欄位如「出版品」、「議題」、「文章」和「MI」為 null 值。由於文件資料庫為半結構化，每個文件都會定義自己的結構，因此這兩個文件在沒有 null 欄位的情況下，可並存於相同文件資料庫中。文件資料庫適合用於處理稀疏資料。

依照文件資料庫進行開發，即可進行快速、重複性的開發。這是因為您可以動態變更文件的資料結構，而不必變更整個集合的結構描述。文件資料庫非常適合靈活的開發工作和動態變化的環境。

了解文件資料庫正規化

文件資料庫不會正規化；在某一個文件中找到的資料可在另一個文件中重複找到。此外，文件之間可能存在一些資料差異。例如，假設您在線上商店購買的情境，而您購買的所有詳細資訊都儲存在單一文件中。此文件可能看起來會像下面的 JSON 文件：

```
{  
  "DateTime": "2018-08-15T12:13:10Z",  
  "LName" : "Santos",  
  "FName" : "Paul",  
  "Cart" : [  
    {  
      "ItemId" : "9876543210123",  
      "Description" : "Understanding Document Databases",  
      "Price" : "29.95"  
    },  
    {  
      "ItemId" : "0123456789012",  
      "Description" : "Programming Today",  
      "Issue": {  
        "Volume": "14",  
        "Number": "09"  
      },  
      "Price" : "8.95"  
    },  
    {  
      "ItemId": "234567890-K",  
      "Description": "Gel Pen (black)",  
      "Price": "2.49"  
    }  
  ],  
}
```

```
"PaymentMethod" :
{
  "Issuer" : "MasterCard",
  "Number" : "1234-5678-9012-3456"
},
"ShopperId" : "1234567890"
}
```

這些資訊全都會做為文件儲存到交易集合中。之後您發現自己忘記購買某一個項目。因此，您再次登入同一家商店並再次購買，這次購買也會存放為交易集合中的另一個文件。

```
{
  "DateTime": "2018-08-15T14:49:00Z",
  "LName" : "Santos",
  "FName" : "Paul",
  "Cart" : [
    {
      "ItemId" : "2109876543210",
      "Description" : "Document Databases for Fun and Profit",
      "Price" : "45.95"
    }
  ],
  "PaymentMethod" :
  {
    "Issuer" : "Visa",
    "Number" : "0987-6543-2109-8765"
  },
  "ShopperId" : "1234567890"
}
```

請注意這兩個文件之間資訊重複的情形您的名稱和顧客 ID (若您使用同一張信用卡，也會有您的信用卡資訊)。但是，這種情況並無不妥，因為儲存空間並不昂貴，而每個文件會完整記錄單一交易，可利用簡單的索引鍵值查詢快速擷取。不需聯結。

這兩個文件之間也存在明顯的差異，即您的信用卡資訊。這只是一項明顯的差異，因為您可能每次購買都使用不同的信用卡。每個文件對於它所記載的交易而言都是正確無誤的。

使用文件

Amazon 文件資料庫是一種文件資料庫，方便儲存、查詢和索引 JSON 資料。在 Amazon DocumentDB 中，集合類似關聯式資料庫中的表格，差別就是不會對所有文件強制採用單一結構描

述。集合可讓您將類似的文件分組，同時將它們全都保存在同一個資料庫中，而不要求文件的結構一致。

若使用前面各節的範例文件，您可能會有 `reading_material` 和 `office_supplies` 的集合。您的軟體需負責強制執行文件屬於哪一個集合。

以下範例使用 MongoDB API 顯示如何新增、查詢、更新和刪除文件。

主題

- [新增文件](#)
- [查詢文件](#)
- [更新文件](#)
- [刪除文件](#)

新增文件

在 Amazon DocumentDB 資料庫中，第一次將文件新增至集合時，會建立資料庫。在此範例中，您在 `test` 資料庫中建立名為 `example` 的集合，這是當您連接到叢集時的預設資料庫。因為插入第一個文件時會隱含地建立集合，所以不會對集合名稱進行錯誤檢查。因此，系列名稱中的錯字 (例如，`eexample` 而不是 `example`) 會建立文件，並將文件新增至 `eexample` 集合，而不是預期的集合。錯誤檢查必須由您的應用程式處理。

以下範例使用 MongoDB API 來新增文件。

主題

- [新增單一文件](#)
- [新增多個文件](#)

新增單一文件

若要新增單一文件至集合，請對您要新增至集合中的文件使用 `insertOne({})` 操作。

```
db.example.insertOne(
  {
    "Item": "Ruler",
    "Colors": ["Red", "Green", "Blue", "Clear", "Yellow"],
    "Inventory": {
      "OnHand": 47,
```



```
        "MinOnHand": 40
    },
    "UnitPrice": 0.89
}
)
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5bedafbcf65ff161707de24f")
}
```

新增多個文件

若要新增多個文件至集合，請對您要新增至集合中的文件清單使用 `insertMany([{}, ..., {}])` 操作。雖然這份特定清單中的文件有不同的結構描述，但是全都可以新增至相同的集合。

```
db.example.insertMany(
  [
    {
      "Item": "Pen",
      "Colors": ["Red", "Green", "Blue", "Black"],
      "Inventory": {
        "OnHand": 244,
        "MinOnHand": 72
      }
    },
    {
      "Item": "Poster Paint",
      "Colors": ["Red", "Green", "Blue", "Black", "White"],
      "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50
      }
    },
    {
      "Item": "Spray Paint",
      "Colors": ["Black", "Red", "Green", "Blue"],
      "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50,
        "OrderQty": 36
      }
    }
  ]
)
```

```
    }  
  }  
]  
)
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "acknowledged" : true,  
  "insertedIds" : [  
    ObjectId("5bedb07941ca8d9198f5934c"),  
    ObjectId("5bedb07941ca8d9198f5934d"),  
    ObjectId("5bedb07941ca8d9198f5934e")  
  ]  
}
```

查詢文件

有時候，您可能需要查詢線上商店的庫存，讓客戶可以查看和購買您銷售的商品。查詢集合相當容易，無論您需要的是集合中的所有文件，或僅需要滿足特定條件的文件。

若要查詢文件，請使用 `find()` 操作。`find()` 命令擁有單一文件參數，負責定義選擇要傳回的文件時所使用的條件。`find()` 輸出的文件格式會是單一行文字且沒有換行。若要將輸出文件格式化，以更方便閱讀，請使用 `find().pretty()`。這個主題的所有範例都使用 `.pretty()` 設定輸出格式。

使用四個文件，可以在 `example` 在前兩個練習中的集合 — `insertOne()` 和 `insertMany()`。

主題

- [擷取集合中的所有文件](#)
- [擷取符合欄位值的文件](#)
- [擷取符合內嵌文件的文件](#)
- [擷取符合內嵌文件中欄位值的文件](#)
- [擷取符合陣列的文件](#)
- [擷取符合陣列中某個值的文件](#)
- [使用運算子擷取文件](#)

擷取集合中的所有文件

若要擷取集合中的所有文件，請使用空的查詢文件使用 `find()` 操作。

以下查詢會傳回 example 集合中的所有文件。

```
db.example.find( {} ).pretty()
```

擷取符合欄位值的文件

若要擷取符合某個欄位和值的所有文件，請使用識別要符合的欄位和值的查詢文件使用 find() 操作。

使用先前文件，此查詢會傳回所有文件，其中「項目」欄位等於「筆」。

```
db.example.find( { "Item": "Pen" } ).pretty()
```

擷取符合內嵌文件的文件

若要尋找符合嵌入文件的所有文件，請使用 find() 操作，其附帶的查詢文件會指定內嵌文件名稱和該內嵌文件的所有欄位和值。

當符合嵌入文件，文件的內嵌文件必須擁有與查詢相同的名稱。此外，內嵌文件的欄位和值必須符合查詢。

以下查詢只會傳回「Poster Paint」(廣告顏料) 文件。這是因為「Pen」(筆) 的 OnHand 和 MinOnHand 有不同值，而「Spray Paint」(噴漆) 比查詢文件多一個欄位 (OrderQty)。

```
db.example.find({"Inventory": {  
  "OnHand": 47,  
  "MinOnHand": 50 } } ).pretty()
```

擷取符合內嵌文件中欄位值的文件

若要尋找符合嵌入文件的所有文件，請使用 find() 操作，其附帶的查詢文件會指定內嵌文件名稱和該內嵌文件的所有欄位和值。

鑒於前述文件，以下查詢使用「點表示法」指定內嵌文件和關注欄位。符合這些的任何文件將會傳回，無論內嵌文件中可能有哪些其他欄位。查詢將傳回「廣告顏料」和「噴漆」，因為它們都符合指定的欄位和值。

```
db.example.find({"Inventory.OnHand": 47, "Inventory.MinOnHand": 50 }).pretty()
```

擷取符合陣列的文件

若要尋找符合陣列的所有文件，請使用 `find()` 操作與您感興趣的陣列名稱，以及在該陣列中的所有值。查詢將傳回所有具有該名稱的文件，其中陣列值與查詢中值及順序相同。

以下查詢只會傳回「筆」，因為「廣告顏料」有額外的顏色 (白色)，而「噴漆」有不同順序的顏色。

```
db.example.find( { "Colors": ["Red","Green","Blue","Black"] } ).pretty()
```

擷取符合陣列中某個值的文件

若要尋找具有特殊陣列值的所有文件，請使用 `find()` 操作與您感興趣的陣列名稱和值。

```
db.example.find( { "Colors": "Red" } ).pretty()
```

前述操作會傳回全部三個文件，因為每個文件都有一個名為 `Colors` 的陣列，且陣列中某處具有「Red」值。如果您指定數值「White」，查詢只會傳回「廣告顏料」。

使用運算子擷取文件

以下查詢會傳回其中「Inventory.OnHand」值小於 50 的所有文件。

```
db.example.find(  
  { "Inventory.OnHand": { $lt: 50 } } )
```

如需支援的查詢運算子清單的詳細資訊，請參閱[查詢和投影運算子](#)。

更新文件

您的文件通常不是靜態的，而且會做為應用程式工作流程的一部分進行更新。以下範例說明一些可更新文件的方式。

若要更新現有文件，請使用 `update()` 操作。`update()` 操作有兩個文件參數。第一份文件識別要進行更新的文件。第二份文件指定要進行的更新。

當您更新現有的欄位，無論該欄位是簡單的欄位、陣列或內嵌文件，只需指定欄位名稱及其值。操作到最後，舊文件的欄位已經由新欄位和值取代。

主題

- [更新現有欄位的值](#)

- [新增新欄位](#)
- [取代與內嵌文件](#)
- [將新欄位加入到內嵌文件](#)
- [從文件中移除欄位](#)
- [從多個文件中刪除欄位](#)

更新現有欄位的值

使用以下您稍早為下列更新操作新增的四個文件。

```
{
  "Item": "Ruler",
  "Colors": ["Red","Green","Blue","Clear","Yellow"],
  "Inventory": {
    "OnHand": 47,
    "MinOnHand": 40
  },
  "UnitPrice": 0.89
},
{
  "Item": "Pen",
  "Colors": ["Red","Green","Blue","Black"],
  "Inventory": {
    "OnHand": 244,
    "MinOnHand": 72
  }
},
{
  "Item": "Poster Paint",
  "Colors": ["Red","Green","Blue","Black","White"],
  "Inventory": {
    "OnHand": 47,
    "MinOnHand": 50
  }
},
{
  "Item": "Spray Paint",
  "Colors": ["Black","Red","Green","Blue"],
  "Inventory": {
    "OnHand": 47,
    "MinOnHand": 50,
```

```
    "OrderQty": 36
  }
}
```

更新簡單欄位

若要更新簡單欄位，請搭配使用 `update()` 和 `$set` 以指定欄位名稱和新值。以下範例會將 `Item` 從「Pen」（筆）變更為「Gel Pen」（簽字筆）。

```
db.example.update(
  { "Item" : "Pen" },
  { $set: { "Item": "Gel Pen" } }
)
```

此操作的結果將會如下所示。

```
{
  "Item": "Gel Pen",
  "Colors": ["Red","Green","Blue","Black"],
  "Inventory": {
    "OnHand": 244,
    "MinOnHand": 72
  }
}
```

更新陣列

以下範例會取代現有顏色陣列和新陣列，其包含 `Orange`，並從顏色清單放棄 `White`。新顏色清單的順序為 `update()` 操作中所指定的順序。

```
db.example.update(
  { "Item" : "Poster Paint" },
  { $set: { "Colors": ["Red","Green","Blue","Orange","Black"] } }
)
```

此操作的結果將會如下所示。

```
{
  "Item": "Poster Paint",
  "Colors": ["Red","Green","Blue","Orange","Black"],
  "Inventory": {
```

```
    "OnHand": 47,  
    "MinOnHand": 50  
  }  
}
```

新增新欄位

若要藉由新增一或多個新的欄位來修改文件，請使用 `update()` 操作搭配查詢文件，以使用 `$set` 運算子識別要插入其中的文件及要插入的新欄位和值。

以下範例會將欄位 `UnitPrice` 的值 3.99 加入到噴漆文件。請注意，值 3.99 是數字，不是字串。

```
db.example.update(  
  { "Item": "Spray Paint" },  
  { $set: { "UnitPrice": 3.99 } }  
)
```

此操作的結果將會如下所示 (JSON 格式)。

```
{  
  "Item": "Spray Paint",  
  "Colors": ["Black","Red","Green","Blue"],  
  "Inventory": {  
    "OnHand": 47,  
    "MinOnHand": 50,  
    "OrderQty": 36  
  },  
  "UnitPrice": 3.99  
}
```

取代與內嵌文件

若要藉由取代內嵌文件來修改文件，請使用 `update()` 操作搭配文件，以使用 `$set` 運算子識別內嵌文件及要插入的新欄位和值。

鑒於以下文件。

```
db.example.insert({  
  "DocName": "Document 1",  
  "Date": {  
    "Year": 1987,
```

```
        "Month": 4,  
        "Day": 18  
    }  
})
```

取代內嵌文件

以下範例將目前日期文件取代為新的欄位，其中只包含 Month 和 Day，Year 已刪除。

```
db.example.update(  
  { "DocName" : "Document 1" },  
  { $set: { "Date": { "Month": 4, "Day": 18 } } }  
)
```

此操作的結果將會如下所示。

```
{  
  "DocName": "Document 1",  
  "Date": {  
    "Month": 4,  
    "Day": 18  
  }  
}
```

將新欄位加入到內嵌文件

將新欄位加入至內嵌文件

若要藉由新增一或多個新欄位至內嵌文件來修改文件，請搭配文件使用 `update()` 操作，其識別內嵌文件及「點表示法」，以使用 `$set` 運算子指定內嵌文件和要插入的新欄位和值。

鑒於以下文件，以下程式碼使用「點表示法」將 Year 和 DoW 欄位插入到內嵌的 Date 文件，並將 Words 插入到父文件。

```
{  
  "DocName": "Document 1",  
  "Date": {  
    "Month": 4,  
    "Day": 18  
  }  
}
```



```
db.example.update(  
  { "DocName" : "Document 1" },  
  { $set: { "Date.Year": 1987,  
           "Date.DoW": "Saturday",  
           "Words": 2482 } }  
)
```

此操作的結果將會如下所示。

```
{  
  "DocName": "Document 1",  
  "Date": {  
    "Month": 4,  
    "Day": 18,  
    "Year": 1987,  
    "DoW": "Saturday"  
  },  
  "Words": 2482  
}
```

從文件中移除欄位

若要藉由移除文件中的欄位來修改文件，請使用 `update()` 操作搭配查詢文件，以使用 `$unset` 運算子識別要從中移除欄位的文件，以指定要移除的欄位。

下列範例會從前述文件中移除 `Words` 欄位。

```
db.example.update(  
  { "DocName" : "Document 1" },  
  { $unset: { Words:1 } }  
)
```

此操作的結果將會如下所示。

```
{  
  "DocName": "Document 1",  
  "Date": {  
    "Month": 4,  
    "Day": 18,  
    "Year": 1987,  
    "DoW": "Saturday"  
  }
```

```
    }  
  }  
}
```

從多個文件中刪除欄位

若要從多個文件中移除欄位來修改文件，請使用 `$unset` 操作搭配 `multi` 運算子，並將 `update()` 選項設定為 `true`。

以下範例會從範例集合的所有文件中移除 `Inventory` 欄位。如果文件沒有 `Inventory` 欄位，則不會對該文件執行任何動作。如果省略 `multi: true`，則只會對符合條件的第一個文件執行動作。

```
db.example.update(  
  {},  
  { $unset: { Inventory:1 } },  
  { multi: true }  
)
```

刪除文件

若要從資料庫中移除文件，請使用 `remove()` 操作，指定要移除的文件。以下程式碼會從您的 `example` 集合移除「簽字筆」。

```
db.example.remove( { "Item": "Gel Pen" } )
```

若要從您的資料庫移除所有文件，請使用 `remove()` 操作與空查詢，如下所示。

```
db.example.remove( { } )
```

開始使用 Amazon DocumentDB

有許多方法可以連接並開始使用 Amazon DocumentDB。我們建立本指南是因為我們發現這種方式是讓使用者開始使用我們強大的文件資料庫的最快、最簡單和最簡單的方式。本指南利用 [AWS Cloud9](#) 用於 Web 的終端機連接和查詢您的 Amazon DocumentDB 叢集使用蒙戈外殼直接從 AWS Management Console 符合免費方案資格的新客戶可以 AWS 免費使用 Amazon DocumentDB。AWS Cloud9 如果您的 AWS Cloud9 環境或 Amazon DocumentDB 叢集使用免費方案以外的資源，則需支付這些資源的正常費 AWS 率。本指南將幫助您在不到 15 分鐘的時間內開始使用 Amazon DocumentDB。

Note

本指南中的指示專門用於建立和連線到以 Amazon DocumentDB 執行個體為基礎的叢集。如果您想要建立並連線到 Amazon DocumentDB 彈性叢集，請參閱 [開始使用 Amazon DocumentDB 彈性叢集](#)

主題

- [必要條件](#)
- [步驟 1：建立 AWS Cloud9 環境](#)
- [步驟 2：建立安全群組](#)
- [第 3 步：創建一個 Amazon DocumentDB 集群](#)
- [第 4 步：安裝蒙戈外殼](#)
- [步驟 5：Connect 到您的 Amazon DocumentDB 集群](#)
- [步驟 6：插入和查詢資料](#)
- [步驟 7：探索](#)

如果您希望透過建立與 Amazon EC2 執行個體的 SSH 連線，從本機電腦連線到 Amazon DocumentDB，請參閱 [使用 EC2 Connect](#) 線說明

必要條件

在建立第一個 Amazon DocumentDB 叢集之前，您必須執行下列動作：

創建一個 Amazon Web Services (AWS) 帳戶

在您可以開始使用 Amazon DocumentDB 之前，您必須有一個 Amazon Web Services (AWS) 帳戶。AWS 帳戶為免費提供。您僅需按使用的服務和資源量付費。

如果您還沒有 AWS 帳戶，請完成以下步驟建立新帳戶。

註冊 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

註冊 AWS 帳戶時，會建立 AWS 帳戶根使用者。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為最佳安全實務，[將管理存取權指派給管理使用者](#)，並且僅使用根使用者來執行[需要根使用者存取權的任務](#)。

設定所需的 AWS Identity and Access Management (IAM) 許可。

存取管理 Amazon DocumentDB 資源 (例如叢集、執行個體和叢集參數群組) 需要 AWS 可用來驗證請求的登入資料。如需詳細資訊，請參閱 [Amazon DocumentDB 的 Identity and Access Management](#)。

1. 在搜尋列中輸入 IAMAWS Management Console，然後在出現的下拉式功能表中選取 IAM。
2. 進入 IAM 主控台後，從導覽窗格中選取 [使用者]。
3. 選擇您的用戶名。
4. 點擊按鈕添加權限。
5. 選取直接連接現有政策。
6. AmazonDocDBFullAccess 在搜索欄中鍵入並在搜索結果中出現時選擇它。
7. 單擊底部顯示「下一步：查看」的藍色按鈕。
8. 點擊底部顯示「添加權限」的藍色按鈕。

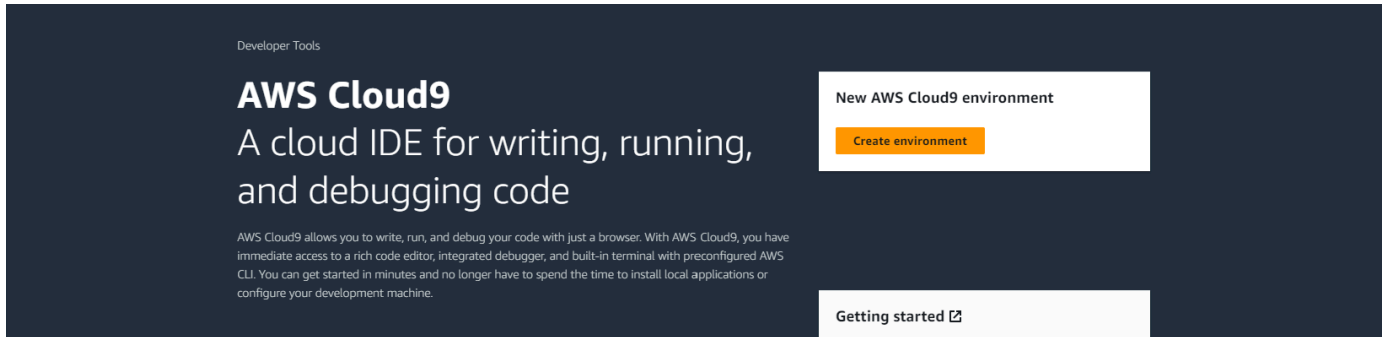
創建一個 Amazon Virtual Private Cloud (Amazon VPC)

只有在您還沒有預設的 Amazon VPC 時，才需要執行此步驟。如果不這樣做，請完成 Amazon VPC 使用者指南中的 [Amazon VPC 入門](#) 步驟 1。這將需要不到五分鐘。

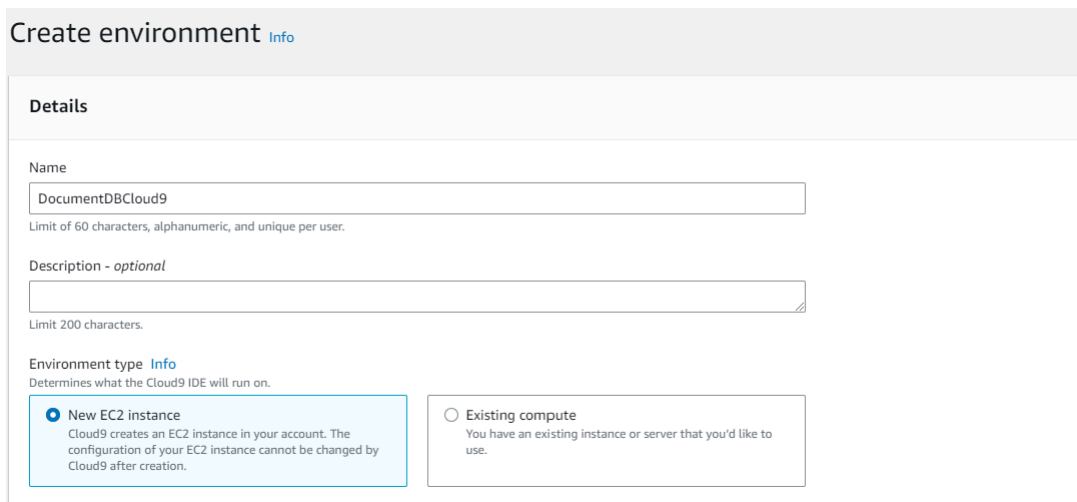
步驟 1：建立AWS Cloud9環境

AWS Cloud9提供一個基於 Web 的終端機，您可以使用該終端機連接到並使用蒙戈外殼查詢您的 Amazon DocumentDB 叢集。

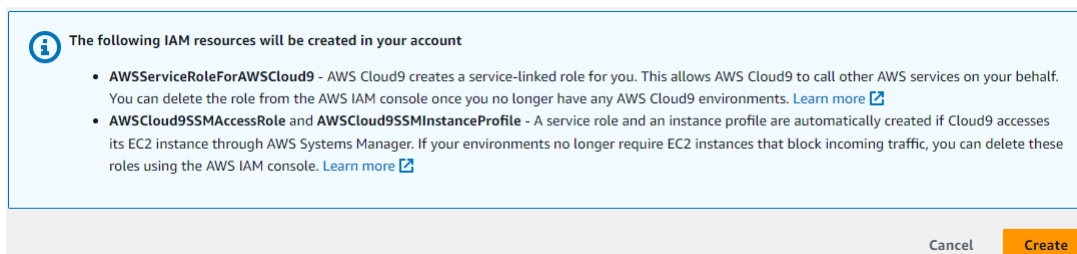
1. 從AWS Management Console導航到AWS Cloud9控制台，然後選擇創建環境。



2. 在「建立環境」對話方塊的「詳細資料」區段DocumentDBCloud9中，於「名稱」欄位中輸入。

The image shows the 'Create environment' dialog box in the AWS Cloud9 console. The 'Name' field is filled with 'DocumentDBCloud9'. Below the name field, there is a description field. The 'Environment type' section shows 'New EC2 instance' selected, with a description: 'Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.' There is also an 'Existing compute' option which is not selected.

3. 對於 [新 EC2 執行個體]、[網路設定] 和 [標籤] 區段，請保留預設設定，然後按一下畫面底部的 [建立]。

The image shows the 'IAM resources' section in the AWS Cloud9 console. It lists the IAM resources that will be created in the account: AWSServiceRoleForAWSCloud9, AWSCloud9SSMAccessRole, and AWSCloud9SSMInstanceProfile. There are 'Cancel' and 'Create' buttons at the bottom right.

您的新AWS Cloud9環境會顯示在「環境」表格中：

| Environments (1) | | | | | | |
|------------------|----------------------|------------------|--------------------|------------|---------------|--|
| Name | Cloud9 IDE | Environment type | Connection | Permission | Owner ARN | |
| DocumentDBCloud9 | Open | EC2 instance | Secure Shell (SSH) | Owner | arn:aws:sts:: | |

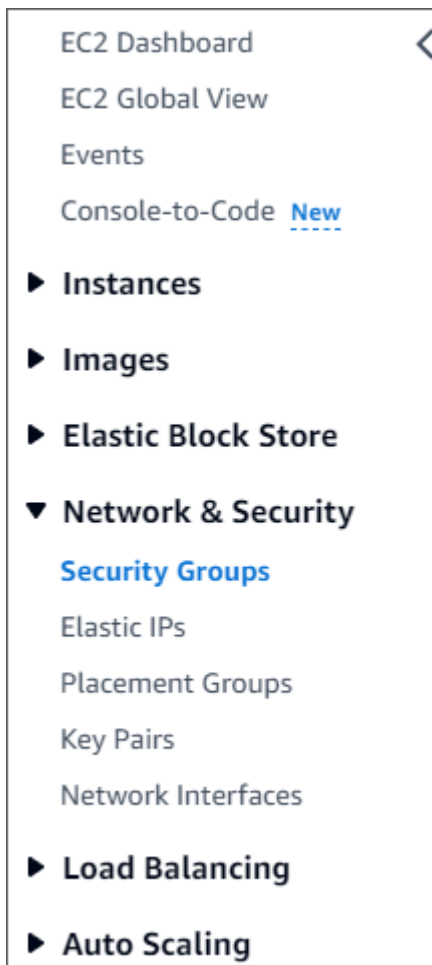
Note

AWS Cloud9環境的佈建最多可能需要三分鐘的時間。

步驟 2：建立安全群組

這個安全群組可讓您從環境連線到 Amazon DocumentDB 叢集。AWS Cloud9

1. 在 [Amazon EC2 管理主控台](#) 的「網路和安全性」下，選擇「安全群組」。



2. 選擇建立安全群組。

Create security group

3. 在「基本詳細資料」區段中：
 - a. 針對 Security group name (安全群組名稱)，輸入 demoDocDB。
 - b. 在描述中，輸入描述。
 - c. 對於 VPC，請接受預設 VPC 的使用。

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

4. 在 Inbound rules (傳入規則) 區段中，選擇 Add rule (新增規則)。
 - a. 針對 Type (類型)，選擇 Custom TCP Rule (自訂 TCP 規則)。
 - b. 對於連接埠範圍，輸入27017。
 - c. 針對來源，選擇您剛建立之AWS Cloud9環境的安全性群組。若要查看可用安全性群組的清單，請cloud9在「來源」欄位右側的搜尋欄位中輸入。選擇名為 `aws-cloud9-<environment name>` 的安全群組。
 - d. 選擇「自訂」做為「目的地」在旁邊的欄位中，搜尋您剛才呼叫的安全性群組demoEC2。您可能需要為 Amazon EC2 主控台重新整理瀏覽器，才能自動填入demoEC2來源名稱。

Inbound rules

| Type | Protocol | Port range | Source | Description - optional |
|------------|----------|------------|--|------------------------|
| Custom TCP | TCP | 27017 | Cust... <input type="text" value="Q"/> | <input type="text"/> |

Note

連接埠 27017 是 Amazon DocumentDB 的預設連接埠。

5. 接受所有其他預設值並選擇建立安全性群組。

Create security group

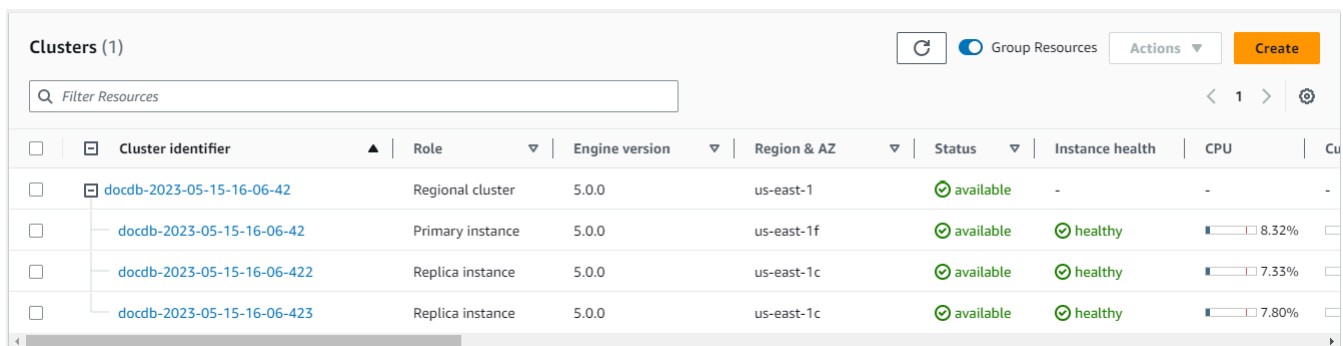
第 3 步：創建一個 Amazon DocumentDB 集群

在此步驟中，您將使用您在上一個步驟中建立的安全群組來建立 Amazon DocumentDB 叢集。

Note

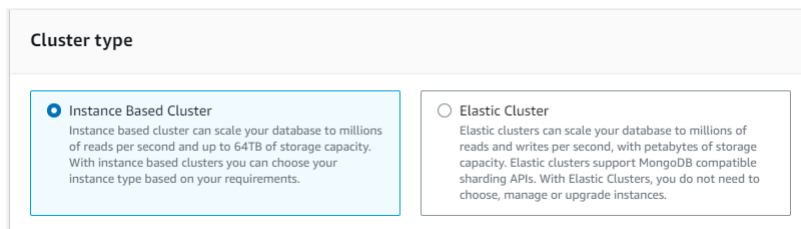
此步驟中的指示專門用於建立以 Amazon DocumentDB 執行個體為基礎的叢集。如果您想要建立 Amazon DocumentDB 彈性叢集，請參閱。[開始使用 Amazon DocumentDB 彈性叢集](#)

1. 在 Amazon DocumentDB 理主控台的叢集下，選擇建立。



| Cluster identifier | Role | Engine version | Region & AZ | Status | Instance health | CPU | Capacity |
|----------------------------|------------------|----------------|-------------|-----------|-----------------|-------|----------|
| docdb-2023-05-15-16-06-42 | Regional cluster | 5.0.0 | us-east-1 | available | - | - | - |
| docdb-2023-05-15-16-06-42 | Primary instance | 5.0.0 | us-east-1f | available | healthy | 8.32% | |
| docdb-2023-05-15-16-06-422 | Replica instance | 5.0.0 | us-east-1c | available | healthy | 7.33% | |
| docdb-2023-05-15-16-06-423 | Replica instance | 5.0.0 | us-east-1c | available | healthy | 7.80% | |

2. 在「建立 Amazon DocumentDB 叢集」頁面的「叢集類型」區段中，選擇「以執行個體為基礎的叢集」（這是預設選項）。

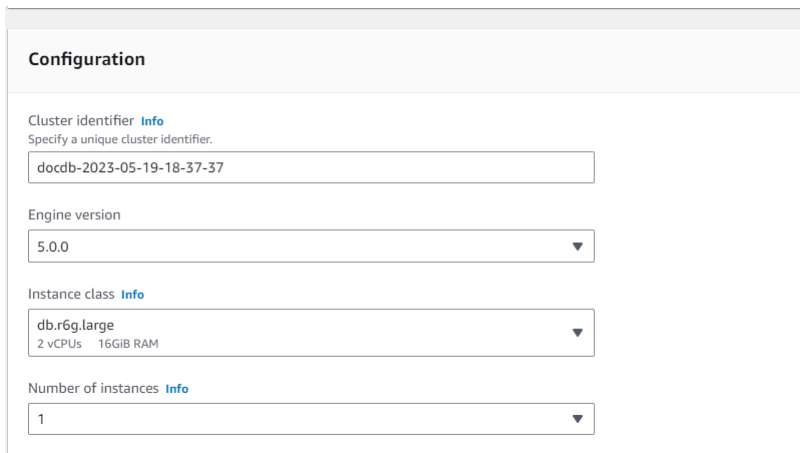


Cluster type

Instance Based Cluster
Instance based cluster can scale your database to millions of reads per second and up to 64TB of storage capacity. With instance based clusters you can choose your instance type based on your requirements.

Elastic Cluster
Elastic clusters can scale your database to millions of reads and writes per second, with petabytes of storage capacity. Elastic clusters support MongoDB compatible sharding APIs. With Elastic Clusters, you do not need to choose, manage or upgrade instances.

3. 在「組態」段落中，選擇 1 個執行處理。選擇一個執行個體有助於降低成本 如果這是生產系統，建議您佈建三個執行個體以獲得高可用性。您可以將「組態」區段中的其他設定保留為預設值。



Configuration

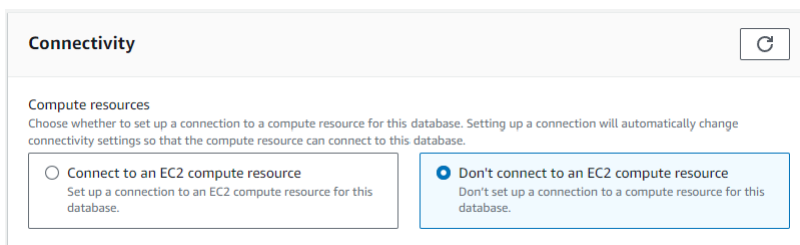
Cluster identifier [Info](#)
Specify a unique cluster identifier.
docdb-2023-05-19-18-37-37

Engine version
5.0.0

Instance class [Info](#)
db.r6g.large
2 vCPUs 16GiB RAM

Number of instances [Info](#)
1

4. 對於連線，請保留預設設定 [不連線至 EC2 運算資源]。



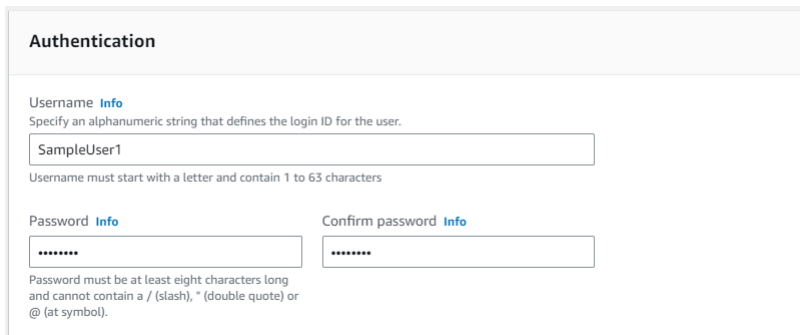
Connectivity

Compute resources
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database.

5. 在「驗證」區段中，輸入登入認證。

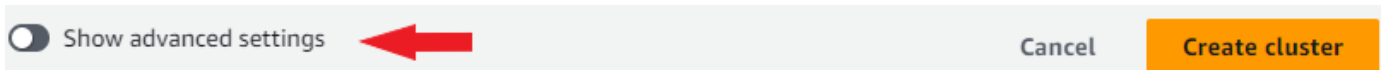



Authentication

Username [Info](#)
Specify an alphanumeric string that defines the login ID for the user.
SampleUser1
Username must start with a letter and contain 1 to 63 characters

Password [Info](#) Confirm password [Info](#)
Password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).

6. 開啟 [顯示進階設定]。



Show advanced settings  Cancel Create cluster

7. 如果您要建立測試或示範叢集，請在 [網路設定] 區段中，針對 VPC 安全性群組選擇 DemoDocDB (VPC)。如果要為生產系統建立叢集，請選擇預設 (VPC)，或者如果要建立特定的 VPC 安全群組，請參閱 Amazon Virtual Private Cloud 使用者指南中的[安全群組](#)。

Network settings

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this cluster.

vpc-02c0445657b77542c

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

Subnet group [Info](#)
A subnet group is a collection of subnets that are within a VPC.

default

VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups

default (VPC) X

8. 選擇 建立叢集。

Show advanced settings
Cancel
Create cluster

Amazon DocumentDB 現在正在佈建您的叢集，最多可能需要幾分鐘的時間才能完成。當叢集和執行個體狀態都顯示為時，您可以連線到叢集 **available**。

i Note

如需叢集狀態值的相關資訊，請參閱《[監控 Amazon DocumentDB](#)》一章 [叢集狀態值](#) 中的。
如需執行個體狀態值的相關資訊，請參閱《[監控 Amazon DocumentDB](#)》一章 [執行個體狀態值](#) 中的。

第 4 步：安裝蒙戈外殼

現在，您將在步驟 1 中創建的 AWS Cloud9 環境中安裝 mongo 外殼。蒙戈殼層是您用來連接和查詢 Amazon DocumentDB 叢集的命令列公用程式。

- 如果您的 AWS Cloud9 環境仍然從步驟 1 開啟，請返回該環境並跳至指令 3。如果您瀏覽離開您的 AWS Cloud9 環境，請在 AWS Cloud9 管理主控台的 [環境] 下，尋找標示為 Document DB Cloud9 的環境。在 Cloud9 IDE 資料行中選擇「開啟」。

| Environments (1) | | | | | | |
|------------------|----------------------|------------------|--------------------|------------|--|--|
| Name | Cloud9 IDE | Environment type | Connection | Permission | Owner ARN | |
| DocumentDBCloud9 | Open | EC2 instance | Secure Shell (SSH) | Owner | arn:aws:sts::713738290397:assumed-role/Admin/michandt-lsengard | |

- 在命令提示字元中，使用下列命令建立儲存庫檔案：

```
echo -e "[mongodb-org-4.0] \nname=MongoDB Repository\nbaseurl=https://\nrepo.mongodb.org/yum/amazon/2013.03/mongodb-org/4.0/x86_64/\nngpgcheck=1 \nenabled=1\nngpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc" | sudo tee /etc/\nyum.repos.d/mongodb-org-4.0.repo
```

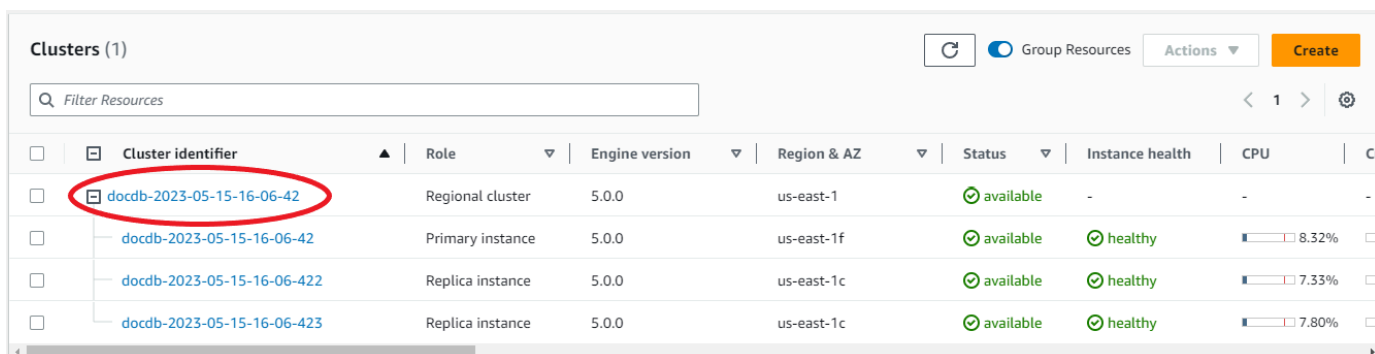
- 完成後，請使用以下命令安裝 mongo 外殼：

```
sudo yum install -y mongodb-org-shell
```

步驟 5：Connect 到您的 Amazon DocumentDB 叢群

現在，您將使用您在步驟 4 中安裝的蒙戈殼層連接到您的 Amazon DocumentDB 叢集。

- 在 Amazon DocumentDB 管理主控台的叢集下，找到您的叢集。按一下叢集識別碼，選擇您建立的叢集。



| Cluster identifier | Role | Engine version | Region & AZ | Status | Instance health | CPU |
|----------------------------------|------------------|----------------|-------------|-----------|-----------------|-------|
| docdb-2023-05-15-16-06-42 | Regional cluster | 5.0.0 | us-east-1 | available | - | - |
| docdb-2023-05-15-16-06-42 | Primary instance | 5.0.0 | us-east-1f | available | healthy | 8.32% |
| docdb-2023-05-15-16-06-422 | Replica instance | 5.0.0 | us-east-1c | available | healthy | 7.33% |
| docdb-2023-05-15-16-06-423 | Replica instance | 5.0.0 | us-east-1c | available | healthy | 7.80% |

- 在默認 encryption-in-transit 情況下在 Amazon DocumentDB 上啟用。您可以選擇性地停用 TLS。若要下載對叢集進行驗證所需的目前憑證，請在「連線與安全性」索引標籤的「連線」區段的「下載對叢集進行驗證所需的 Amazon DocumentDB 憑證授權單位 (CA) 憑證」下，複製提供的 Connect 字串。返回您的 AWS Cloud9 環境並粘貼連接字符串。

Connectivity & security | Instances | Configuration | Monitoring | Events & tags | Maintenance & backups

Connect

Getting Started Guide | Enabling/Disabling TLS | Connecting programmatically

Download the Amazon DocumentDB Certificate Authority (CA) certificate required to authenticate to your cluster [Copy](#)

```
wget https://truststore.pki.rds.amazonaws.com/us-east-1/us-east-1-bundle.pem
```

Connect to this cluster with the mongo shell [Copy](#)

```
mongo --ssl --host docdb-2023-06-02-14-26-22.cluster-cozt4xr9xv9b.us-east-1.docdb.amazonaws.com:27017 --sslCAFile us-east-1-bundle.pem --username testuser --password <insertYourPassword>
```

Connect to this cluster with an application [Copy](#)

```
mongodb://testuser:<insertYourPassword>@docdb-2023-06-02-14-26-22.cluster-cozt4xr9xv9b.us-east-1.docdb.amazonaws.com:27017/?ssl=true&ssl_ca_certs=us-east-1-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false
```

- 在 Amazon DocumentDB 主控台的連線與安全性索引標籤的「連線」區段的「使用 mongo 殼層連線到此叢集」下，返回叢集，複製提供的 Connect 字串。省略複製，<insertYourPassword>以便在連接時由 mongo shell 提示您輸入密碼。

Connectivity & security | Instances | Configuration | Monitoring | Events & tags | Maintenance & backups

Connect

Getting Started Guide | Enabling/Disabling TLS | Connecting programmatically

Download the Amazon DocumentDB Certificate Authority (CA) certificate required to authenticate to your cluster [Copy](#)

```
wget https://truststore.pki.rds.amazonaws.com/us-east-1/us-east-1-bundle.pem
```

Connect to this cluster with the mongo shell [Copy](#)

```
mongo --ssl --host docdb-2023-06-02-14-26-22.cluster-cozt4xr9xv9b.us-east-1.docdb.amazonaws.com:27017 --sslCAFile us-east-1-bundle.pem --username testuser --password <insertYourPassword>
```

Connect to this cluster with an application [Copy](#)

```
mongodb://testuser:<insertYourPassword>@docdb-2023-06-02-14-26-22.cluster-cozt4xr9xv9b.us-east-1.docdb.amazonaws.com:27017/?ssl=true&ssl_ca_certs=us-east-1-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false
```

返回您的AWS Cloud9環境並粘貼連接字符串。

當您輸入密碼且提示出現提示時，即表rs0:PRIMARY>示您已成功連線至 Amazon DocumentDB 叢集。

Note

如需疑難排解的相關資訊，請參閱[疑難排解 Amazon DocumentDB](#)。

步驟 6：插入和查詢資料

現在您已連線到叢集，您可以執行一些查詢以熟悉文件資料庫的使用。

1. 若要插入單一文件，請輸入下列內容：

```
db.collection.insert({"hello":"DocumentDB"})
```

2. 您會得到以下輸出：

```
WriteResult({ "nInserted" : 1 })
```

3. 您可以讀取使用findOne()指令撰寫的文件 (因為它只會傳回單一文件)。輸入以下內容：

```
db.collection.findOne()
```

4. 您會得到以下輸出：

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" : "DocumentDB"
  }
```

5. 若要執行更多查詢，請考慮使用遊戲設定檔使用案例。首先，將幾個項目插入到標題為的集合中profiles。輸入以下內容：

```
db.profiles.insertMany([
  { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
    "score":202},
  { "_id" : 2, "name" : "Frank", "status": "inactive", "level":
    2, "score":9},
  { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
    "score":87},
  { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
    "score":27}
])
```

6. 您會得到以下輸出：

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. 使用指find()令傳回縱斷面集合中的所有文件。輸入以下內容：

```
db.profiles.find()
```

8. 您將獲得與您在步驟 5 中輸入的數據匹配的輸出。
9. 使用篩選器查詢單一文件。輸入以下內容：

```
db.profiles.find({name: "Katie"})
```

10. 你應該得到這個輸出：

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
  "score":27}
```

11. 現在，讓我們嘗試查找配置文件並使用findAndModify命令對其進行修改。我們將使用以下代碼給用戶 Matt 額外的十分：

```
db.profiles.findAndModify({
  query: { name: "Matt", status: "active"},
  update: { $inc: { score: 10 } }
})
```

12. 您會得到以下輸出（請注意，他的分數尚未增加）：

```
{
  "_id" : 1,
  "name" : "Matt",
  "status" : "active",
  "level" : 12,
  "score" : 202
}
```

13. 您可以通過以下查詢驗證他的分數是否已更改：

```
db.profiles.find({name: "Matt"})
```

14. 您會得到以下輸出：

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12, "score"
  : 212 }
```

步驟 7：探索

恭喜您！您已成功完成 Amazon DocumentDB 的入門指南。

下一步是什麼？了解如何充分利用這個數據庫的一些熱門功能：

- [管理 Amazon DocumentDB](#)
- [擴展](#)
- [備份與還原](#)

Note

除非您將其刪除，否則您從此入門練習建立的叢集將繼續累積成本。如需指示，請參閱[刪除 Amazon DocumentDB 叢集](#)。

Amazon DocumentDB 快速入門使用 AWS CloudFormation

本節包含步驟和其他資訊，可協助您快速開始使用 Amazon DocumentDB (具有 MongoDB 相容性)。 [AWS CloudFormation](#) 如需有關亞馬遜文件資料庫的一般資訊，請參閱 [什麼是 Amazon DocumentDB \(與 MongoDB 相容性\)](#)。

這些指示使用 AWS CloudFormation 範本在預設 Amazon VPC 中建立叢集和執行個體。如需自行建立這些資源的說明，請參閱 [開始使用 Amazon DocumentDB](#)。

Important

由此範本建立的 AWS CloudFormation 堆疊會建立多個資源，包括 Amazon DocumentDB 中的資源 (例如叢集和執行個體) 和 Amazon 彈性運算雲端 (例如子網路群組) 中的資源。其中部分資源不屬於免費方案資源。如需定價資訊，請參閱 [Amazon DocumentDB 定價](#) 和 [Amazon EC2 定價](#)。當您完成後可以刪除堆疊，以停止計費。

此 AWS CloudFormation 堆疊僅供教學課程使用。如果您在生產環境中使用此範本，建議您使用更嚴格的 IAM 政策和安全性。如需保護資源安全的相關資訊，請參閱 [Amazon VPC 安全性](#) 和 [Amazon EC2 網路和安全性](#)。

主題

- [必要條件](#)
- [啟動 Amazon DocumentDB 堆棧 AWS CloudFormation](#)
- [存取 Amazon DocumentDB 叢集](#)
- [終止保護及刪除保護](#)

必要條件

在建立 Amazon DocumentDB 叢集之前，您必須具備下列項目：

- 一個默認的 Amazon VPC
- 必要的 IAM 許可

所需的 IAM 許可

以下權限可讓您建立 AWS CloudFormation 堆疊的資源：

AWS受管理策略

- `AWSCloudFormationReadOnlyAccess`
- `AmazonDocDBFullAccess`

其他 IAM 許可

下列政策說明了建立和刪除此 AWS CloudFormation 堆疊所需的額外許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetSSHPublicKey",
        "iam:ListSSHPublicKeys",
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:PutRolePolicy",
        "iam:CreateInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:GetAccountSummary",
        "iam:ListAccountAliases",
        "iam:GetRole",
        "iam:DeleteRole",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:DeleteRolePolicy",
        "iam:DeleteInstanceProfile",
        "cloudformation:*Stack",
        "ec2:DescribeKeyPairs",
        "ec2:*Vpc",
        "ec2:DescribeInternetGateways",
        "ec2:*InternetGateway",
        "ec2:createTags",
        "ec2:*VpcAttribute",
        "ec2:DescribeRouteTables",
        "ec2:*RouteTable",
```

```

        "ec2:*Subnet",
        "ec2:*SecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:DescribeVpcEndpoints",
        "ec2:*VpcEndpoint",
        "ec2:*SubnetAttribute",
        "ec2:*Route",
        "ec2:*Instances",
        "ec2:DeleteVpcEndpoints"
    ],
    "Resource": "*"
},
{
    "Sid": "iamPassRole",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": "rds.amazonaws.com"
        }
    }
}
]
}
}

```

Note

在前述政策中的粗體許可僅在刪除堆疊時為必

要：`iam>DeleteRole`、`iam:RemoveRoleFromInstanceProfile`、`iam>DeleteRolePolicy`、`iam>DeleteRolePermissionsBoundary` 和 `ec2:DeleteVpcEndpoints`。亦請注意 `ec2:*Vpc` 授予 `ec2:DeleteVpc` 許可。

Amazon EC2 金鑰對

您要建立 AWS CloudFormation 堆疊的區域中必須有可用的金鑰對 (和 PEM 檔案)。如果您需要建立金鑰對，請參閱 [Amazon EC2 執行個體使用者指南中的使用 Amazon EC2 建立 key pair](#)。

啟動 Amazon DocumentDB 堆棧 AWS CloudFormation

本節說明如何啟動和設定 Amazon DocumentDB AWS CloudFormation 堆疊。

1. 前往 <https://console.aws.amazon.com/> 登入 AWS Management Console。
2. 下表列出每個範本的 Amazon DocumentDB 堆疊範本。AWS 區域為AWS 區域您想要啟動堆疊的選擇啟動堆疊。

| 區域 | 檢視範本 | 在設計工具中檢視 | 啟動 |
|----------------|----------------------|--------------------------|---|
| 美國東部 (俄亥俄) | 檢視範本 | 在設計工具中檢視 |  |
| 美國東部 (維吉尼亞北部) | 檢視範本 | 在設計工具中檢視 |  |
| 美國西部 (奧勒岡) | 檢視範本 | 在設計工具中檢視 |  |
| 亞太區域 (孟買) | 檢視範本 | 在設計工具中檢視 |  |
| 亞太區域 (首爾) | 檢視範本 | 在設計工具中檢視 |  |
| 亞太區域 (新加坡) | 檢視範本 | 在設計工具中檢視 |  |
| 亞太區域 (雪梨) | 檢視範本 | 在設計工具中檢視 |  |
| 亞太區域 (東京) | 檢視範本 | 在設計工具中檢視 |  |
| 加拿大 (中部) | 檢視範本 | 在設計工具中檢視 |  |
| 歐洲 (法蘭克福) | 檢視範本 | 在設計工具中檢視 |  |
| 歐洲 (愛爾蘭) | 檢視範本 | 在設計工具中檢視 |  |
| 歐洲 (倫敦) | 檢視範本 | 在設計工具中檢視 |  |
| Europe (Paris) | 檢視範本 | 在設計工具中檢視 |  |

3. 建立堆疊 — 說明您選取的 Amazon DocumentDB 範本。每個堆疊都是以 JSON 或 YAML 檔案的範本為基礎，其中包含您要包含在堆疊中之AWS資源的相關設定。因為您選擇從上述提供的範本啟動堆疊，因此您的範本已設定為為AWS 區域您選擇的範本建立 Amazon DocumentDB 堆疊。

當您啟動AWS CloudFormation堆疊時，預設會停用 Amazon DocumentDB 叢集的[刪除保護](#)。如果您要啟用叢集的刪除保護，請完成以下步驟。否則，請選擇 Next (下一步) 以繼續下一個步驟。

若要為您的 Amazon DocumentDB 叢集啟用刪除保護：

1. 從 [建立堆疊] 頁面右下角選擇 [在設計工具中檢視]。
2. 在主控台的結果 AWS CloudFormation 設計工具頁面中，使用整合的 JSON 和 YAML 編輯器來修改範本。捲動至 Resources 區段並修改它以包含 DeletionProtection，如下所示。如需使用 AWS CloudFormation 設計工具的詳細資訊，請參閱[什麼是 AWS CloudFormation 設計工具？](#)。

JSON：

```
"Resources": {
  "DBCluster": {
    "Type": "AWS::DocDB::DBCluster",
    "DeletionPolicy": "Delete",
    "Properties": {
      "DBClusterIdentifier": {
        "Ref": "DBClusterName"
      },
      "MasterUsername": {
        "Ref": "MasterUser"
      },
      "MasterUserPassword": {
        "Ref": "MasterPassword"
      },
      "DeletionProtection": "true"
    }
  },
}
```

YAML：

```
Resources:
  DBCluster:
    Type: 'AWS::DocDB::DBCluster'
    DeletionPolicy: Delete
```

```
Properties:
  DBClusterIdentifier: !Ref DBClusterName
  MasterUsername: !Ref MasterUser
  MasterUserPassword: !Ref MasterPassword
  DeletionProtection: 'true'
```

3. 選擇頁面左上角的 Create Stack (建立堆疊)



以儲存變更，並在啟用這些變更的情況下建立堆疊。

4. 儲存變更之後，系統會將您重新導向至 Create stack (建立堆疊) 頁面。

5. 選擇 Next (下一步) 繼續。

4. 指定堆疊詳細資訊 — 輸入範本的堆疊名稱和參數。參數是在您的範本中定義，而且可讓您在建立或更新堆疊時輸入自訂值。

- 在 Stack name (堆疊名稱) 下，輸入堆疊的名稱或接受提供的名稱。堆疊名稱可以包含字母 (A—Z 和 a—z)、數字 (0—9) 和破折號 (—)。
- 在 Parameters (參數) 下，輸入下列詳細資訊：
 - 資料庫 ClusterName — 輸入 Amazon DocumentDB 叢集的名稱，或接受提供的名稱。

叢集命名限制條件：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- Amazon RDS、Neptune 和亞馬 Amazon DocumentDB 中每AWS 帳戶個區域的所有叢集必須是唯一的。
- 資料庫 InstanceClass — 從下拉式清單中，選取 Amazon DocumentDB 叢集的執行個體類別。
- 資料庫 InstanceName — 輸入 Amazon DocumentDB 執行個體的名稱，或接受提供的名稱。

執行個體命名限制：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

- Amazon RDS、Neptune 和 Amazon DocumentDB 中每AWS 帳戶個區域的所有執行個體必須是唯一的。
- MasterPassword— 資料庫管理員帳戶密碼。
- MasterUser— 資料庫管理員帳戶使用者名稱。 MasterUser 必須以字母開頭，且只能包含英數字元。

選擇 Next (下一步) 儲存變更並繼續。

5. 設定堆疊選項 — 設定堆疊的標籤、權限和其他選項。

- 標籤 — 指定要套用至堆疊中資源的標籤 (鍵值) 配對。您最多可以為每個堆疊新增 50 個唯一的標籤。
- 權限 — 選用。選擇 IAM 角色以明確定義如AWS CloudFormation何建立、修改或刪除堆疊中的資源。如果您沒有選擇角色，AWS CloudFormation 會根據您的使用者認證使用權限。在您指定服務角色之前，請確保您有傳遞許可 (iam:PassRole)。iam:PassRole 許可會指定您可使用哪些角色。

Note

當您指定某個服務角色時，AWS CloudFormation 一律使用該角色處理在該堆疊上執行的所有操作。其他具有在此堆疊上執行這些操作之許可的使用者，都能夠使用此角色，即使他們不具傳遞它的許可。如果該角色包含使用者不該有的許可，您可能在無意中提升使用者的許可。確定該角色授予[最低權限](#)。

- 進階選項 — 您可以設定下列進階選項：
 - 堆疊原則 — 選用。可藉由此選項來定義堆疊更新期間，您欲避免意外更新的資源。在預設情況下，系統可以在堆疊更新期間將所有資源進行更新。

您可以直接輸入堆疊政策作為 JSON，或上傳包含堆疊政策的 JSON 檔案。如需詳細資訊，請參閱[避免更新堆疊資源](#)。

- 回復配置 — 可選。指定建立和更新堆疊時AWS CloudFormation要監視的 CloudWatch 記錄警示。如果作業違反警示閾值，請將其 AWS CloudFormation 轉返。
- 通知選項 — 可選。指定簡易通知系統 (SNS) 的主題。
- 堆疊建立選項 — 選用。您可以指定下列選項：
 - 失敗時復原 — 如果堆疊建立失敗，是否應該回復堆疊。
 - 逾時 — 堆疊建立逾時前的分鐘數。
 - 終止保護 — 防止意外刪除堆疊。

Note

AWS CloudFormation 終止保護 Amazon DocumentDB 刪除保護的概念不同。如需詳細資訊，請參閱 [終止保護及刪除保護](#)。

選擇 Next (下一步) 繼續。

6. 檢閱 <stack-name>— 檢閱您的堆疊範本、詳細資料和組態選項。您也可以在頁面底部開啟快速建立連結，以建立與此相同基本組態的堆疊。
 - 選擇 Create (建立) 以建立堆疊。
 - 或者，您可以選擇 Create change set (建立變更集)。變更集是建立堆疊之前如何配置此堆疊的預覽。這可讓您在執行變更集之前檢查各種模型組態。

存取 Amazon DocumentDB 叢集

AWS CloudFormation 堆疊完成後，您可以使用 Amazon EC2 執行個體連線到您的 Amazon DocumentDB 叢集。如需使用安全殼層連線至 Amazon EC2 執行個體的相關資訊，請參閱 [Amazon EC2 Linux 執行個體使用者指南中的 Connect 到您的 Linux 執行個體](#)。

連線之後，請參閱下列各節，其中包含使用 Amazon DocumentDB 的相關資訊。

- [第 4 步：安裝蒙戈外殼](#)
- [刪除 Amazon DocumentDB 集群](#)

終止保護及刪除保護

啟用刪除保護和終止保護是 Amazon DocumentDB 的最佳實務。CloudFormation 終止保護與 Amazon DocumentDB 刪除保護功能截然不同的功能。

- 終止保護 — 您可以透過為堆疊啟用終止保護來避免意外刪除 CloudFormation 堆疊。如果使用者嘗試刪除啟用終止保護的堆疊，則刪除會失敗，且堆疊保持不變。當您使用建立堆疊時，依預設會停用終止保護 CloudFormation。您可以在建立新的堆疊時啟用終止保護。如需更多資訊，請參閱 [設定 AWS CloudFormation 堆疊選項](#)。
- 刪除保護 — Amazon DocumentDB 也提供針對叢集啟用刪除保護的功能。如果使用者嘗試刪除已啟用刪除保護的 Amazon DocumentDB 叢集，則刪除作業會失敗，且叢集保持不變。刪除保

護在啟用時，可防止意外刪除 Amazon DocumentDB AWS Management Console、AWS CLI和 CloudFormation 如需啟用和停用 Amazon DocumentDB 叢集之刪除保護的詳細資訊，請參閱。[刪除保護](#)

MongoDB 性

Amazon DocumentDB 支持 MongoDB 的兼容性，包括 MongoDB 4.0 和 MongoDB 5.0。MongoDB 相容性表示您目前已經與 MongoDB 資料庫搭配使用的絕大多數應用程式、驅動程式和工具都可以與 Amazon DocumentDB 搭配使用，而不需要變更很少或不需要變更。本節說明 Amazon DocumentDB 與 MongoDB 相容性所需瞭解的所有資訊，包括新功能和功能、入門、移轉路徑和功能差異。

主題

- [MongoDB 性](#)
- [MongoDB 4.0 兼容性](#)

MongoDB 性

主題

- [Amazon DocumentDB 5.0 中的新功能](#)
- [開始使用 Amazon DocumentDB 5.0](#)
- [升級或遷移到 Amazon DocumentDB 4.0](#)
- [功能差異](#)

Amazon DocumentDB 5.0 中的新功能

Amazon DocumentDB 5.0 引入了新功能，包括儲存限制和用戶端欄位層級加密。下面的摘要介紹了亞馬遜 5.0 中引入的一些主要功能。若要查看新權能的完整清單，請參閱[版本備註](#)。

- 對於所有以執行個體為基礎的 Amazon DocumentDB 叢集和以資料為基礎的彈性叢集，將儲存限制增加到 128 TiB。
- 推出 Amazon DocumentDB 5.0 引擎版本 3.0.775)
 - Support MongoDB 動程式
 - Support 用戶端欄位層級加密 (FLE)。您現在可以在將資料寫入 Amazon DocumentDB 叢集之前，先在用戶端加密欄位。如需詳細資訊，請參閱[用戶端欄位層級加密](#)
 - 新的聚合運算符：`$dateAdd`和`$dateSubtract`
 - 支持與`$elemMatch`運算符索引。因此，具有的查詢`$elemMatch`將導致索引掃描。

Amazon DocumentDB 不支持每個 MongoDB 5.0 功能。當我們建置 Amazon DocumentDB 5.0 時，我們會從客戶要求我們建置最多的功能和功能向後運作。我們將繼續根據客戶要求我們構建的內容添加其他 MongoDB 5.0 功能。如需支援 API 的最新清單，請參閱[支援的 MongoDB API、營運和資料類型](#)。

開始使用 Amazon DocumentDB 5.0

若要開始使用 Amazon DocumentDB 5.0，請參閱[入門](#)指南。您可以使用或 AWS 開發套件，AWS Management Console AWS CLI 或建立新的 Amazon 文件資料庫 5.0 叢集。AWS CloudFormation 連接到 Amazon DocumentDB 時，需要您使用與 MongoDB 5.0 或更高版本兼容的 MongoDB 驅動程序或實用程序。

Note

使用 AWS SDK、或 AWS CloudFormation 時 AWS CLI，引擎版本將預設為 5.0.0。您必須明確指定參數 `engineVersion = 4.0.0` 才能建立新的 Amazon DocumentDB 4.0 叢集或建立新 `engineVersion = 3.6.0` 的 Amazon DocumentDB 3.6 叢集。對於指定的 Amazon DocumentDB 叢集，您可以使用呼叫 AWS CLI 來判斷叢集版本，`describe-db-clusters` 或使用 Amazon DocumentDB 管理主控台來檢視特定叢集的引擎版本號碼。

Amazon DocumentDB 5.0 支援 Amazon EC2 重力 2 處理器 (例如和叢集的 `t4.medium` 執行個體類型)，`r6g` 並且可在所有支援的區域使用。如需有關定價的詳細資訊，請參閱 [Amazon DocumentDB \(與 MongoDB 相容性\) 定價](#)。

升級或遷移到 Amazon DocumentDB 4.0

您可以從 MongoDB 3.6 或 MongoDB 4.0 遷移到 Amazon DocumentDB 5.0 使用 [AWS DMS](#) 或實用程序 `mongodump`，`mongoexport` 和 `mongoimport` `mongoexport` 如需有關如何移轉的指示，請參閱 [使用升級您的 Amazon DocumentDB 集群 AWS Database Migration Service](#)。

功能差異

Amazon DocumentDB 4.0 和 5.0 之間的功能差異

隨著 Amazon DocumentDB 5.0 的發布，有 Amazon DocumentDB 3.6 和 Amazon DocumentDB 4.0 之間的功能差異：

- 備份內建角色現在支援 `serverStatus`。動作-具備備份角色的開發人員和應用程式可以收集有關 Amazon DocumentDB 叢集狀態的統計資料。

- SecondaryDelaySecs欄位會slaveDelay在replSetGetConfig輸出中取代。
- 此命hello令取代 isMaster-hello 傳回描述 Amazon DocumentDB 叢集角色的文件。
- Amazon DocumentDB 5.0 現在支援使用第一個巢狀層級的\$elemMatch操作員進行索引掃描。如果僅查詢篩選具有一個層級的篩選器，則支援索引掃描，但如果包含巢狀\$elemMatch查詢，則不支援索引掃描。\$elemMatch

例如，在 Amazon DocumentDB 5.0，如果你包括在嵌套級別的\$elemMatch運算符，它不會返回一個值，因為它在 Amazon DocumentDB 4.0：

```
db.foo.insert(
[
  {a: {b: 5}},
  {a: {b: [5]}},
  {a: {b: [3, 7]}},
  {a: [{b: 5}]},
  {a: [{b: 3}, {b: 7}]},
  {a: [{b: [5]}]},
  {a: [{b: [3, 7]}]},
  {a: [[{b: 5}]}],
  {a: [[{b: 3}, {b: 7}]]},
  {a: [[{b: [5]}]]},
  {a: [[{b: [3, 7]}]]}
]);

// DocumentDB 5.0
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }

// DocumentDB 4.0
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }
{ "a" : [ [ { "b" : [ 5 ] } ] ] }
```

- Amazon DocumentDB 4.0 中的「\$」投影返回所有字段的所有文檔。使用 Amazon DocumentDB 5.0 時，具有「\$」投影的find命令會傳回符合僅包含符合「\$」投影欄位的查詢參數的文件。
- 在 Amazon DocumentDB 5.0 中，具有\$regex和\$options查詢參數的find命令返回一個錯誤：「無法同時在\$regex和\$options中設置選項」。
- 使用 Amazon DocumentDB 5.0，\$indexOfCP現在在以下情況返回「-1」：
 - 在字符串表達式中找不到子字符串，或者
 - 開始是一個大於結束的數字，或

- `start` 是一個大於字符串字節長度的數字。
- 在 Amazon DocumentDB 4.0, `$indexOfCP` 返回「0」當開始位置是一個數字大於結束或字符串的字節長度。
- 例如 `{"_id.nestedField" : 1}`, 使用 Amazon DocumentDB 5.0 時, 投影操作會傳回僅包含預計欄位 `_id` `fields` 的文件。而在 Amazon DocumentDB 4.0 中, 嵌套的字段投影命令不會過濾掉任何文檔。

MongoDB 4.0 兼容性

主題

- [Amazon DocumentDB 4.0 功能](#)
- [開始使用 Amazon DocumentDB 4.0](#)
- [升級或遷移到 Amazon DocumentDB 4.0](#)
- [功能差異](#)

Amazon DocumentDB 4.0 功能

Amazon DocumentDB 4.0 引入了許多新功能和功能, 其中包括 ACID 交易以及對變更串流的改進。下面的摘要顯示了 Amazon DocumentDB 4.0 中引入的一些主要功能。若要查看權能的完整清單, 請參閱 [版本備註](#)。

- **ACID 交易** : Amazon DocumentDB 現在支援跨多個文件、陳述式、集合和資料庫執行交易的能力。交易可讓您在 Amazon DocumentDB 叢集中的一或多個文件執行原子、一致、隔離且耐用 (ACID) 作業, 以簡化應用程式開發。如需詳細資訊, 請參閱 [交易](#)。
- **變更串流** : 您現在可以在叢集層級 (`client.watch()` 或 `mongo.watch()`) 和 Database (`db.watch()`) 開啟變更串流, 您可以指定開啟變更串流游標, 最後您現在可以將變更串流保留期延長至 7 天 (之前 24 小時)。 `startAtOperationTime` 如需詳細資訊, 請參閱 [使用更改流與亞馬遜文檔數據庫](#)。
- **AWS Database Migration Service (AWS DMS)** : 您現在可以使用 AWS DMS 將 MongoDB 4.0 工作負載遷移到 Amazon DocumentDB。 AWS DMS 現在支持 MongoDB 4.0 源, Amazon DocumentDB 4.0 目標和 Amazon DocumentDB 3.6 源執行 Amazon DocumentDB 3.6 和 4.0 之間的升級。如需詳細資訊, 請參閱 [AWS DMS 文件](#)。
- **效能和索引** : 您現在可以使用索引 `$lookup`, 尋找包含一個欄位或一個欄位的投影查詢, 而且可以直接從索引提供 `_id` 欄位, 而不需要從集合中讀取 (涵蓋的查詢)、使 `hint()` 用的功

能 `findAndModify`、效能最佳化 `$addToSet`，以及減少整體索引大小的改進。如需詳細資訊，請參閱 [版本備註](#)。

- **運算子**：Amazon DocumentDB 4.0 現在支援一些新的彙總運算子：`$ifNull`、`$replaceRoot`、`$setIsSubset`、`$setIntersection`、`$setUnion`。`$se` 可以看到所有的 MongoDB 的 API，操作和數據類型，我們支援在 [支援的 MongoDB API、營運和資料類型](#)。
- **以角色為基礎的存取控制 (RBAC)**：使用 `ListCollection` 和 `ListDatabase` 命令，您現在可以選擇性地使用 `authorizedCollections` 和 `authorizedDatabases` 參數，讓使用者列出他們有權存取的集合和資料庫，而不需要 `listCollections` 和 `listDatabase` 角色。您也可以在不需 `KillCursor` 角色的情況下殺死自己的游標。

Amazon DocumentDB 不支持每個 MongoDB 4.0 功能。當我們建置 Amazon DocumentDB 4.0 時，我們會從客戶要求我們建置最多的功能和功能向後執行。我們將繼續根據客戶要求我們構建的內容添加其他 MongoDB 4.0 功能。例如，Amazon DocumentDB 4.0 目前不支持類型轉換運算符或在 MongoDB 4.0 中引入的字符串運算符。如需支援 API 的最新清單，請參閱 [支援的 MongoDB API、營運和資料類型](#)。

開始使用 Amazon DocumentDB 4.0

若要開始使用 Amazon DocumentDB 4.0，請參閱 [入門](#) 指南。您可以使用或 AWS 開發套件、AWS Management Console AWS CLI 或建立新的 Amazon 文件資料庫 4.0 叢集。AWS CloudFormation 連接到 Amazon DocumentDB 時，需要您使用與 MongoDB 4.0 或更高版本兼容的 MongoDB 驅動程序或實用程序。

Note

使用 AWS SDK、或 AWS CloudFormation 時 AWS CLI，引擎版本將預設為 5.0.0。您必須明確指定參數 `engineVersion = 4.0.0` 才能建立新的 Amazon DocumentDB 4.0 叢集或建立新 `engineVersion = 3.6.0` 的 Amazon DocumentDB 3.6 叢集。對於指定的 Amazon DocumentDB 叢集，您可以使用呼叫 AWS CLI 來判斷叢集版本，`describe-db-clusters` 或使用 Amazon DocumentDB 管理主控台來檢視特定叢集的引擎版本號碼。

Amazon DocumentDB 4.0 為您的叢集支援 `r5r6gt3.medium`、和 `t4g.medium` 執行個體類型，並且可在所有受支援的區域中使用。使用 Amazon DocumentDB 4.0 無需額外費用。如需有關定價的詳細資訊，請參閱 [Amazon DocumentDB \(與 MongoDB 相容性\) 定價](#)。

升級或遷移到 Amazon DocumentDB 4.0

您可以從 MongoDB 3.6 或 MongoDB 4.0 遷移到 Amazon DocumentDB 4.0 使用 [AWS DMS](#) 或公用事業 [mongodump](#)，如 [，mongoexport](#) 和 [mongoimport](#) 同樣，您可以使用相同的工具從 Amazon DocumentDB 3.6 升級到 Amazon DocumentDB 4.0。如需有關如何移轉的指示，請參閱 [使用升級您的 Amazon DocumentDB 集群 AWS Database Migration Service](#)。

功能差異

Amazon DocumentDB 3.6 和 4.0 之間的功能差異

隨著 Amazon DocumentDB 4.0 的發布，有 Amazon DocumentDB 3.6 和 Amazon DocumentDB 4.0 之間的功能差異：

- 巢狀文件的投影：Amazon DocumentDB 3.6 會在套用投影時考慮巢狀文件中的第一個欄位。但是，Amazon DocumentDB 4.0 將解析子文檔，並將投影應用於每個子文檔以及。例如：如果投影是 "a.b.c": 1，則兩個版本中的行為是相同的。但是，如果投影是 {a:{b:{c:1}}} 那麼 Amazon DocumentDB 3.6 將只適用於投影到 'a' 而不是 'B' 或 'c'。
- 行為 **minKey** **maxKey**：在 Amazon DocumentDB 4.0 中，行為不會 {x:{\$gt:MaxKey}} 返回任何內容，並且返回所有內容。{x:{\$lt:MaxKey}}
- 文件比較差異：比較子文件 (例如 in) 中不同類型 (double、int、long) 的數值現在可 b 在 {"_id" :1, "a" : {"b":1}} 數值資料類型與文件的每個層級之間提供一致的輸出。

Amazon DocumentDB 4.0 和 MongoDB 數據庫 4.0 之間的功能差異

以下是 Amazon DocumentDB 4.0 和 MongoDB 4.0 之間的功能差異。

- 在路徑中使用空鍵查找：當集合包含在數組內部具有空鍵的文檔 (例如 {"x" : [{ "" : 10 }, { "b" : 20 }]})，並且當查詢中使用的鍵以空字符串 (例如 x.) 結束時，Amazon DocumentDB 將返回該文檔，因為它遍歷數組中的所有文檔，而 MongoDB 不會返回該文檔。
- **\$setOnInsert** 隨著 \$ 在路徑中：字段運算符 \$setOnInsert 將不會與 \$ 在亞馬遜 DocumentDB，這也與 MongoDB 4.0 一致的路徑組合作。

交易

Amazon DocumentDB (to MongoDB 兼容性) 現在支援 MongoDB 4.0 (MongoDB)，包括交易。您可以跨多個文件、陳述式、集合和資料庫執行交易。交易可讓您在 Amazon DocumentDB 叢集中的一或多個文件執行原子、一致、隔離且耐用 (ACID) 作業，以簡化應用程式開發。交易的常見使用案例包括財務處理、履行和管理訂單，以及建立多人遊戲。

交易無需額外付費。您只需為作為交易一部分使用的讀寫 iOS 付費。

主題

- [要求](#)
- [最佳實務](#)
- [限制](#)
- [監控與診斷](#)
- [交易隔離層級](#)
- [使用案例](#)
- [支援的命令](#)
- [不支援](#)
- [工作階段](#)
- [交易錯誤](#)

要求

若要使用交易功能，您必須符合下列需求：

- 您必須使用 Amazon DocumentDB 4.0 引擎。
- 您必須使用與 MongoDB 4.0 或更高版本相容的驅動程式。

最佳實務

以下是一些最佳實務，讓您可以充分利用 Amazon DocumentDB 的交易。

- 一律在交易完成後提交或中止交易。讓交易處於不完整的狀態會繫結資料庫資源，並可能導致寫入衝突。

- 建議將事務保持在所需的命令的最小數量。如果您有具有多個陳述式的交易，這些陳述式可以分成多個較小的交易，建議您這樣做，以減少逾時的可能性。始終致力於創建短期交易，而不是長時間運行的讀取。

限制

- Amazon DocumentDB 不支持交易中的游標。
- Amazon DocumentDB 無法在交易中創建新的集合，也無法對不存在的集合進行查詢/更新。
- 文件層級寫入鎖定會受到 1 分鐘逾時的影響，使用者無法設定此逾時。
- Amazon DocumentDB 不支援可重試寫入、可重試的提交和可重試中止命令。例外：如果您使用的是 mongo shell，請不要在任何代碼字符串中包含該 `retryWrites=false` 命令。根據預設，可重試寫入都會停用。包括 `retryWrites=false` 可能會導致正常讀取命令失敗。
- 每個 Amazon DocumentDB 執行個體在執行個體上同時開啟的並行交易數都有上限。有關限制，請參閱[執行個體限制](#)。
- 對於指定的交易，交易記錄檔大小必須小於 32MB。
- Amazon DocumentDB 在交易 `count()` 中確實支援，但並非所有驅動程式都支援此功能。另一種方法是使用 `countDocuments()` API，該 API 將計數查詢轉換為客戶端的聚合查詢。
- 交易有一分鐘的執行限制，工作階段有 30 分鐘逾時。如果事務超時，它將被中止，並且在現有事務的會話內發出的任何後續命令將產生以下錯誤：

```
WriteCommandError({
  "ok" : 0,
  "operationTime" : Timestamp(1603491424, 627726),
  "code" : 251,
  "errmsg" : "Given transaction number 0 does not match any in-progress transactions."
})
```

監控與診斷

透過 Amazon DocumentDB 4.0 中的交易支援，我們新增了額外的 CloudWatch 指標來協助您監控交易。

新 CloudWatch 量度

- `DatabaseTransactions`：在一分鐘期間內取得的未結交易數目。

- DatabaseTransactionsAborted：在一分鐘期間執行的中止交易數目。
- DatabaseTransactionsMax：一分鐘期間內未結交易的最大數量。
- TransactionsAborted：在一分鐘期間內，執行個體中止的交易數目。
- TransactionsCommitted：一分鐘期間內，執行處理確認的交易數目。
- TransactionsOpen：在一分鐘期間執行處理開啟的交易數目。
- TransactionsOpenMax：一分鐘期間內，執行處理開啟的交易數目上限。
- TransactionsStarted：在一分鐘期間內在執行個體上開始的交易數目。

Note

如需 Amazon DocumentDB 的更多 CloudWatch 指標，請移至[使用監控亞馬遜文檔數據庫 CloudWatch](#)。

此外，新欄位已新增至currentOpIpsidtransactionThreadId、，以及「idle transaction」和serverStatus交易的新狀態：currentActivecurrentInactive、currentOpen、totalAborted、totalCommitted、和totalStarted。

交易隔離層級

開始交易時，您可以指定readConcern和，writeConcern如下面的範例所示：

```
mySession.startTransaction({readConcern: {level: 'snapshot'}, writeConcern: {w: 'majority'}});
```

對於readConcern，Amazon DocumentDB 預設支援快照隔離。如果指定readConcern了本機、可用或大多數，Amazon DocumentDB 會將層級升readConcern級為快照。Amazon DocumentDB 不支援線性化，而readConcern且指定這類讀取問題將導致錯誤。

對於writeConcern，Amazon DocumentDB 預設支援多數資料庫，而且在三個 AZ 中保留四個資料副本時，就會達到寫入仲裁。如果指定了writeConcern較低的，Amazon DocumentDB 將升級writeConcern到多數。此外，所有 Amazon DocumentDB 寫入都會以日誌記錄，而且無法停用日誌記錄。

使用案例

在本節中，我們將逐步介紹兩個交易用例：多語句和多收集。

多重對帳單交易

Amazon DocumentDB 交易是多重陳述式，這表示您可以透過明確提交或復原來撰寫跨越多個陳述式的交易。您可以將insert、updatedelete、和findAndModify動作分組為單一原子作業。

多重對帳單交易的常見使用案例是借記-貸方交易。例如：你欠一個朋友錢買衣服。因此，您需要從您的帳戶中借記（提取）500 美元，並將 \$500（存款）存入您朋友的帳戶。若要執行該作業，您可以在單一交易中同時執行債務和貸方作業，以確保原子性。這樣做可以防止 500 美元從您的帳戶中扣除，但不會記入您朋友的帳戶的情況。這是這個用例將是什麼樣子：

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success Scenario***
// Setup bank account for Alice and Bob. Each have $1000 in their account

var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"}, {"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountColl.update({"name": "Bob"}, {"$set": {"balance": newBobBalance}});
```

```
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;

session.commitTransaction();

accountColl.find();

// *** Transfer $500 from Alice to Bob inside a transaction: Failure Scenario***

// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"}, {"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;

session.abortTransaction();
```

多重收集異動

我們的交易也是多集合，這意味著它們可以用於在單個事務中和跨多個集合執行多個操作。這可提供一致的資料檢視，並維護資料的完整性。當您將命令提交為單個命令時<>，事務將被 all-or-nothing 執行-在此，它們將全部成功或全部失敗。

以下是多集合交易的範例，使用多重陳述式交易範例中的相同案例和資料。

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success Scenario***
```

```
// Setup bank account for Alice and Bob. Each have $1000 in their account
var amountToTransfer = 500;
var collectionName = "account";

var session = db.getMongo().startSession({causalConsistency: false});
var accountCollInBankA = session.getDatabase("bankA")[collectionName];
var accountCollInBankB = session.getDatabase("bankB")[collectionName];

accountCollInBankA.drop();
accountCollInBankB.drop();

accountCollInBankA.insert({name: "Alice", balance: 1000});
accountCollInBankB.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountCollInBankA.update({"name": "Alice"}, {"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountCollInBankB.update({"name": "Bob"}, {"$set": {"balance": newBobBalance}});
var findBobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;

session.commitTransaction();

accountCollInBankA.find(); // Alice holds $500 in bankA
accountCollInBankB.find(); // Bob holds $1500 in bankB

// *** Transfer $500 from Alice to Bob inside a transaction: Failure Scenario***

// Setup bank account for Alice and Bob. Each have $1000 in their account
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var accountCollInBankA = session.getDatabase("bankA")[collectionName];
var accountCollInBankB = session.getDatabase("bankB")[collectionName];

accountCollInBankA.drop();
```

```
accountCollInBankB.drop();

accountCollInBankA.insert({name: "Alice", balance: 1000});
accountCollInBankB.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountCollInBankA.update({"name": "Alice"}, {"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountCollInBankB.update({"name": "Bob"}, {"$set": {"balance": newBobBalance}});
var findBobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;

session.abortTransaction();

accountCollInBankA.find(); // Alice holds $1000 in bankA
accountCollInBankB.find(); // Bob holds $1000 in bankB
```

回呼 API 的交易 API 範例

回調 API 僅適用於 4.2 + 驅動程序。

Javascript

下面的代碼演示如何使用 Amazon DocumentDB 交易 API 與 JavaScript。

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success ***
// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();
```

```
accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert(aliceBalance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"}, {"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert.eq(newAliceBalance, findAliceBalance);

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountColl.update({"name": "Bob"}, {"$set": {"balance": newBobBalance}});
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;
assert.eq(newBobBalance, findBobBalance);

session.commitTransaction();

accountColl.find();
```

Node.js

下列程式碼示範如何利用與 Node.js 的 Amazon DocumentDB 交易 API。

```
// Node.js callback API:

const bankDB = await MongoClient.db("bank");
var accountColl = await bankDB.createCollection("account");
var amountToTransfer = 500;

const session = MongoClient.startSession({causalConsistency: false});
await accountColl.drop();

await accountColl.insertOne({name: "Alice", balance: 1000}, { session });
await accountColl.insertOne({name: "Bob", balance: 1000}, { session });

const transactionOptions = {
  readConcern: { level: 'snapshot' },
  writeConcern: { w: 'majority' }
```

```
};

// deduct $500 from Alice's account
var aliceBalance = await accountColl.findOne({name: "Alice"}, {session});
assert(aliceBalance.balance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
session.startTransaction(transactionOptions);
await accountColl.updateOne({name: "Alice"}, {$set: {balance: newAliceBalance}},
    {session });
await session.commitTransaction();
aliceBalance = await accountColl.findOne({name: "Alice"}, {session});
assert(newAliceBalance == aliceBalance.balance);

// add $500 to Bob's account
var bobBalance = await accountColl.findOne({name: "Bob"}, {session});
var newBobBalance = bobBalance.balance + amountToTransfer;
session.startTransaction(transactionOptions);
await accountColl.updateOne({name: "Bob"}, {$set: {balance: newBobBalance}},
    {session });
await session.commitTransaction();
bobBalance = await accountColl.findOne({name: "Bob"}, {session});
assert(newBobBalance == bobBalance.balance);
```

C#

下列程式碼示範如何搭配 C# 使用 Amazon DocumentDB 交易 API。

```
// C# Callback API

var dbName = "bank";
var collName = "account";
var amountToTransfer = 500;

using (var session = client.StartSession(new ClientSessionOptions{CausalConsistency
    = false}))
{
    var bankDB = client.GetDatabase(dbName);
    var accountColl = bankDB.GetCollection<BsonDocument>(collName);
    bankDB.DropCollection(collName);
    accountColl.InsertOne(session, new BsonDocument { {"name", "Alice"}, {"balance",
    1000 } });
    accountColl.InsertOne(session, new BsonDocument { {"name", "Bob"}, {"balance",
    1000 } });
}
```

```
// start transaction
var transactionOptions = new TransactionOptions(
    readConcern: ReadConcern.Snapshot,
    writeConcern: WriteConcern.WMajority);
var result = session.WithTransaction(
    (sess, cancellationtoken) =>
    {
        // deduct $500 from Alice's account
        var aliceBalance = accountColl.Find(sess,
Builders<BsonDocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
        Debug.Assert(aliceBalance >= amountToTransfer);
        var newAliceBalance = aliceBalance.AsInt32 - amountToTransfer;
        accountColl.UpdateOne(sess, Builders<BsonDocument>.Filter.Eq("name",
"Alice"),
                                Builders<BsonDocument>.Update.Set("balance",
newAliceBalance));
        aliceBalance = accountColl.Find(sess,
Builders<BsonDocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
        Debug.Assert(aliceBalance == newAliceBalance);

        // add $500 from Bob's account
        var bobBalance = accountColl.Find(sess,
Builders<BsonDocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
        var newBobBalance = bobBalance.AsInt32 + amountToTransfer;
        accountColl.UpdateOne(sess, Builders<BsonDocument>.Filter.Eq("name",
"Bob"),
                                Builders<BsonDocument>.Update.Set("balance",
newBobBalance));
        bobBalance = accountColl.Find(sess,
Builders<BsonDocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
        Debug.Assert(bobBalance == newBobBalance);

        return "Transaction committed";
    }, transactionOptions);
// check values outside of transaction
var aliceNewBalance = accountColl.Find(Builders<BsonDocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
var bobNewBalance = accountColl.Find(Builders<BsonDocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
Debug.Assert(aliceNewBalance == 500);
```



```
    Debug.Assert(bobNewBalance == 1500);
}
```

Ruby

下列程式碼示範如何搭配紅寶石使用 Amazon DocumentDB 交易 API。

```
// Ruby Callback API

dbName = "bank"
collName = "account"
amountToTransfer = 500

session = client.start_session(:causal_consistency=> false)
bankDB = Mongo::Database.new(client, dbName)
accountColl = bankDB[collName]
accountColl.drop()

accountColl.insert_one({"name"=>"Alice", "balance"=>1000})
accountColl.insert_one({"name"=>"Bob", "balance"=>1000})

# start transaction
session.with_transaction(read_concern: {level: :snapshot}, write_concern:
{w: :majority}) do
  # deduct $500 from Alice's account
  aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
session).first['balance']
  assert aliceBalance >= amountToTransfer
  newAliceBalance = aliceBalance - amountToTransfer
  accountColl.update_one({"name"=>"Alice"}, { "$set" =>
{"balance"=>newAliceBalance} }, :session=> session)
  aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
session).first['balance']
  assert_equal(newAliceBalance, aliceBalance)

  # add $500 from Bob's account
  bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
session).first['balance']
  newBobBalance = bobBalance + amountToTransfer
  accountColl.update_one({"name"=>"Bob"}, { "$set" =>
{"balance"=>newBobBalance} }, :session=> session)
  bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
session).first['balance']
  assert_equal(newBobBalance, bobBalance)
```

```
end

# check results outside of transaction
aliceBalance = accountColl.find({"name"=>"Alice"}).first['balance']
bobBalance = accountColl.find({"name"=>"Bob"}).first['balance']
assert_equal(aliceBalance, 500)
assert_equal(bobBalance, 1500)

session.end_session
```

Go

下列程式碼示範如何搭配 Go 使用 Amazon DocumentDB 交易 API。

```
// Go - Callback API
type Account struct {
    Name string
    Balance int
}

ctx := context.TODO()

dbName := "bank"
collName := "account"
amountToTransfer := 500

session, err := client.StartSession(options.Session().SetCausalConsistency(false))
assert.NoError(t, err)
defer session.EndSession(ctx)

bankDB := client.Database(dbName)
accountColl := bankDB.Collection(collName)
accountColl.Drop(ctx)

_, err = accountColl.InsertOne(ctx, bson.M{"name" : "Alice", "balance":1000})
_, err = accountColl.InsertOne(ctx, bson.M{"name" : "Bob", "balance":1000})

transactionOptions := options.Transaction().SetReadConcern(readconcern.Snapshot()).
    SetWriteConcern(writeconcern.New(writeconcern.WMajority()))
_, err = session.WithTransaction(ctx, func(sessionCtx mongo.SessionContext)
(interface{}, error) {
    var result Account
    // deduct $500 from Alice's account
```

```
err = accountColl.FindOne(sessionCtx, bson.M{"name": "Alice"}).Decode(&result)
aliceBalance := result.Balance
newAliceBalance := aliceBalance - amountToTransfer
_, err = accountColl.UpdateOne(sessionCtx, bson.M{"name": "Alice"},
bson.M{"$set": bson.M{"balance": newAliceBalance}})
err = accountColl.FindOne(sessionCtx, bson.M{"name": "Alice"}).Decode(&result)
aliceBalance = result.Balance
assert.Equal(t, aliceBalance, newAliceBalance)

// add $500 to Bob's account
err = accountColl.FindOne(sessionCtx, bson.M{"name": "Bob"}).Decode(&result)
bobBalance := result.Balance
newBobBalance := bobBalance + amountToTransfer
_, err = accountColl.UpdateOne(sessionCtx, bson.M{"name": "Bob"}, bson.M{"$set":
bson.M{"balance": newBobBalance}})
err = accountColl.FindOne(sessionCtx, bson.M{"name": "Bob"}).Decode(&result)
bobBalance = result.Balance
assert.Equal(t, bobBalance, newBobBalance)

if err != nil {
    return nil, err
}
return "transaction committed", err
}, transactionOptions)

// check results outside of transaction
var result Account
err = accountColl.FindOne(ctx, bson.M{"name": "Alice"}).Decode(&result)
aliceNewBalance := result.Balance
err = accountColl.FindOne(ctx, bson.M{"name": "Bob"}).Decode(&result)
bobNewBalance := result.Balance
assert.Equal(t, aliceNewBalance, 500)
assert.Equal(t, bobNewBalance, 1500)
// Go - Core API
type Account struct {
    Name string
    Balance int
}

func transferMoneyWithRetry(sessionContext mongo.SessionContext, accountColl
*mongo.Collection, t *testing.T) error {
    amountToTransfer := 500
```

```
    transactionOptions :=
options.Transaction().SetReadConcern(readconcern.Snapshot()).

SetWriteConcern(writeconcern.New(writeconcern.WMajority()))
    if err := sessionContext.StartTransaction(transactionOptions); err != nil {
        panic(err)
    }

    var result Account
    // deduct $500 from Alice's account
    err := accountColl.FindOne(sessionContext, bson.M{"name":
"Alice"}).Decode(&result)
    aliceBalance := result.Balance
    newAliceBalance := aliceBalance - amountToTransfer
    _, err = accountColl.UpdateOne(sessionContext, bson.M{"name": "Alice"},
bson.M{"$set": bson.M{"balance": newAliceBalance}})
    if err != nil {
        sessionContext.AbortTransaction(sessionContext)
    }
    err = accountColl.FindOne(sessionContext, bson.M{"name":
"Alice"}).Decode(&result)
    aliceBalance = result.Balance
    assert.Equal(t, aliceBalance, newAliceBalance)

    // add $500 to Bob's account
    err = accountColl.FindOne(sessionContext, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance := result.Balance
    newBobBalance := bobBalance + amountToTransfer
    _, err = accountColl.UpdateOne(sessionContext, bson.M{"name": "Bob"},
bson.M{"$set": bson.M{"balance": newBobBalance}})
    if err != nil {
        sessionContext.AbortTransaction(sessionContext)
    }
    err = accountColl.FindOne(sessionContext, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance = result.Balance
    assert.Equal(t, bobBalance, newBobBalance)

    err = sessionContext.CommitTransaction(sessionContext)
    return err
}

func doTransactionWithRetry(t *testing.T) {
    ctx := context.TODO()
```

```

    dbName := "bank"
    collName := "account"
    bankDB := client.Database(dbName)
    accountColl := bankDB.Collection(collName)

    client.UseSessionWithOptions(ctx, options.Session().SetCausalConsistency(false),
func(sessionContext mongo.SessionContext) error {
    accountColl.Drop(ctx)
    accountColl.InsertOne(sessionContext, bson.M{"name" : "Alice",
"balance":1000})
    accountColl.InsertOne(sessionContext, bson.M{"name" : "Bob",
"balance":1000})
    for {
        err := transferMoneyWithRetry(sessionContext, accountColl, t)
        if err == nil {
            println("transaction committed")
            return nil
        }
        if mongoErr := err.(mongo.CommandError);
mongoErr.HasErrorLabel("TransientTransactionError") {
            continue
        }
        println("transaction failed")
        return err
    }
})

// check results outside of transaction
var result Account
accountColl.FindOne(ctx, bson.M{"name": "Alice"}).Decode(&result)
aliceBalance := result.Balance
assert.Equal(t, aliceBalance, 500)
accountColl.FindOne(ctx, bson.M{"name": "Bob"}).Decode(&result)
bobBalance := result.Balance
assert.Equal(t, bobBalance, 1500)
}

```

Java

下面的代碼演示如何利用 Amazon DocumentDB 交易 API 與 Java。

```

// Java (sync) - Callback API
MongoDatabase bankDB = mongoClient.getDatabase("bank");
MongoCollection accountColl = bankDB.getCollection("account");

```

```
accountColl.drop();
int amountToTransfer = 500;

// add sample data
accountColl.insertOne(new Document("name", "Alice").append("balance", 1000));
accountColl.insertOne(new Document("name", "Bob").append("balance", 1000));

TransactionOptions txnOptions = TransactionOptions.builder()
    .readConcern(ReadConcern.SNAPSHOT)
    .writeConcern(WriteConcern.MAJORITY)
    .build();
ClientSessionOptions sessionOptions =
    ClientSessionOptions.builder().causallyConsistent(false).build();
try ( ClientSession clientSession = mongoClient.startSession(sessionOptions) ) {
    clientSession.withTransaction(new TransactionBody<Void>() {
        @Override
        public Void execute() {
            // deduct $500 from Alice's account
            List<Document> documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
"Alice")).into(documentList);
            int aliceBalance = (int) documentList.get(0).get("balance");
            int newAliceBalance = aliceBalance - amountToTransfer;

            accountColl.updateOne(clientSession, new Document("name", "Alice"), new
Document("$set", new Document("balance", newAliceBalance)));

            // check Alice's new balance
            documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
"Alice")).into(documentList);
            int updatedBalance = (int) documentList.get(0).get("balance");
            Assert.assertEquals(updatedBalance, newAliceBalance);

            // add $500 to Bob's account
            documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
"Bob")).into(documentList);
            int bobBalance = (int) documentList.get(0).get("balance");
            int newBobBalance = bobBalance + amountToTransfer;

            accountColl.updateOne(clientSession, new Document("name", "Bob"), new
Document("$set", new Document("balance", newBobBalance)));
        }
    });
}
```

```
        // check Bob's new balance
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
"Bob")).into(documentList);
        updatedBalance = (int) documentList.get(0).get("balance");
        Assert.assertEquals(updatedBalance, newBobBalance);

        return null;
    }
}, txnOptions);
}
```

C

下列程式碼示範如何搭配 C 使用 Amazon DocumentDB 交易 API。

```
// Sample Code for C with Callback

#include <bson.h>
#include <mongoc.h>
#include <stdio.h>
#include <string.h>
#include <assert.h>

typedef struct {
    int64_t balance;
    bson_t *account;
    bson_t *opts;
    mongoc_collection_t *collection;
} ctx_t;

bool callback_session (mongoc_client_session_t *session, void *ctx, bson_t **reply,
    bson_error_t *error)
{
    bool r = true;
    ctx_t *data = (ctx_t *) ctx;
    bson_t local_reply;
    bson_t *selector = data->account;
    bson_t *update = BCON_NEW ("$set", "{", "balance", BCON_INT64 (data->balance),
    "}");

    mongoc_collection_update_one (data->collection, selector, update, data->opts,
    &local_reply, error);
}
```

```
*reply = bson_copy (&local_reply);
bson_destroy (&local_reply);
bson_destroy (update);
return r;
}

void test_callback_money_transfer(mongoc_client_t* client, mongoc_collection_t*
collection, int amount_to_transfer){

    bson_t reply;
    bool r = true;
    const bson_t *doc;
    bson_iter_t iter;
    ctx_t alice_ctx;
    ctx_t bob_ctx;
    bson_error_t error;

    // find query
    bson_t *alice_query = bson_new ();
    BSON_APPEND_UTF8(alice_query, "name", "Alice");

    bson_t *bob_query = bson_new ();
    BSON_APPEND_UTF8(bob_query, "name", "Bob");

    // create session
    // set causal consistency to false
    mongoc_session_opt_t *session_opts = mongoc_session_opts_new ();
    mongoc_session_opts_set_causal_consistency (session_opts, false);
    // start the session
    mongoc_client_session_t *client_session = mongoc_client_start_session (client,
session_opts, &error);

    // add session to options
    bson_t *opts = bson_new();
    mongoc_client_session_append (client_session, opts, &error);

    // deduct 500 from Alice
    // find account balance of Alice
    mongoc_cursor_t *cursor = mongoc_collection_find_with_opts (collection,
alice_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t alice_balance = (bson_iter_value (&iter))->value.v_int64;
```



```
assert(alice_balance >= amount_to_transfer);
int64_t new_alice_balance = alice_balance - amount_to_transfer;

// set variables which will be used by callback function
alice_ctx.collection = collection;
alice_ctx.opts = opts;
alice_ctx.balance = new_alice_balance;
alice_ctx.account = alice_query;

// callback
r = mongoc_client_session_with_transaction (client_session, &callback_session,
NULL, &alice_ctx, &reply, &error);
assert(r);

// find account balance of Alice after transaction
cursor = mongoc_collection_find_with_opts (collection, alice_query, NULL, NULL);
mongoc_cursor_next (cursor, &doc);
bson_iter_init (&iter, doc);
bson_iter_find (&iter, "balance");
alice_balance = (bson_iter_value (&iter))->value.v_int64;
assert(alice_balance == new_alice_balance);
assert(alice_balance == 500);

    // add 500 to bob's balance
// find account balance of Bob
cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
mongoc_cursor_next (cursor, &doc);
bson_iter_init (&iter, doc);
bson_iter_find (&iter, "balance");
int64_t bob_balance = (bson_iter_value (&iter))->value.v_int64;
int64_t new_bob_balance = bob_balance + amount_to_transfer;

bob_ctx.collection = collection;
bob_ctx.opts = opts;
bob_ctx.balance = new_bob_balance;
bob_ctx.account = bob_query;

// set read & write concern
mongoc_read_concern_t *read_concern = mongoc_read_concern_new ();
mongoc_write_concern_t *write_concern = mongoc_write_concern_new ();
mongoc_transaction_opt_t *txn_opts = mongoc_transaction_opts_new ();

mongoc_write_concern_set_w(write_concern, MONGOC_WRITE_CONCERN_W_MAJORITY);
mongoc_read_concern_set_level(read_concern, MONGOC_READ_CONCERN_LEVEL_SNAPSHOT);
```

```
mongoc_transaction_opts_set_write_concern (txn_opts, write_concern);
mongoc_transaction_opts_set_read_concern (txn_opts, read_concern);

// callback
r = mongoc_client_session_with_transaction (client_session, &callback_session,
txn_opts, &bob_ctx, &reply, &error);
assert(r);

// find account balance of Bob after transaction
cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
mongoc_cursor_next (cursor, &doc);
bson_iter_init (&iter, doc);
bson_iter_find (&iter, "balance");
bob_balance = (bson_iter_value (&iter))->value.v_int64;
assert(bob_balance == new_bob_balance);
assert(bob_balance == 1500);

// cleanup
bson_destroy(alice_query);
bson_destroy(bob_query);
mongoc_client_session_destroy(client_session);
bson_destroy(opts);
mongoc_transaction_opts_destroy(txn_opts);
mongoc_read_concern_destroy(read_concern);
mongoc_write_concern_destroy(write_concern);
mongoc_cursor_destroy(cursor);
bson_destroy(doc);
}
int main(int argc, char* argv[]) {
    mongoc_init ();
    mongoc_client_t* client = mongoc_client_new (<connection uri>);
    bson_error_t error;

    // connect to bank db
    mongoc_database_t *database = mongoc_client_get_database (client, "bank");
    // access account collection
    mongoc_collection_t* collection = mongoc_client_get_collection(client, "bank",
"account");
    // set amount to transfer
    int64_t amount_to_transfer = 500;
    // delete the collection if already existing
    mongoc_collection_drop(collection, &error);

    // open Alice account
```

```
bson_t *alice_account = bson_new ();
BSON_APPEND_UTF8(alice_account, "name", "Alice");
BSON_APPEND_INT64(alice_account, "balance", 1000);

// open Bob account
bson_t *bob_account = bson_new ();
BSON_APPEND_UTF8(bob_account, "name", "Bob");
BSON_APPEND_INT64(bob_account, "balance", 1000);

bool r = true;

r = mongoc_collection_insert_one(collection, alice_account, NULL, NULL, &error);
if (!r) {printf("Error encountered:%s", error.message);}
r = mongoc_collection_insert_one(collection, bob_account, NULL, NULL, &error);
if (!r) {printf("Error encountered:%s", error.message);}

test_callback_money_transfer(client, collection, amount_to_transfer);

}
```

Python

下面的代碼演示如何利用 Amazon DocumentDB 交易 API 與 Python。

```
// Sample Python code with callback api

import pymongo

def callback(session, balance, query):
    collection.update_one(query, {'$set': {"balance": balance}}, session=session)

client = pymongo.MongoClient(<connection uri>)
rc_snapshot = pymongo.read_concern.ReadConcern('snapshot')
wc_majority = pymongo.write_concern.WriteConcern('majority')

# To start, drop and create an account collection and insert balances for both Alice
  and Bob
collection = client.get_database("bank").get_collection("account")
collection.drop()
collection.insert_one({"_id": 1, "name": "Alice", "balance": 1000})
collection.insert_one({"_id": 2, "name": "Bob", "balance": 1000})

amount_to_transfer = 500
```

```
# deduct 500 from Alice's account
alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert alice_balance >= amount_to_transfer
new_alice_balance = alice_balance - amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.with_transaction(lambda s: callback(s, new_alice_balance, {"name":
        "Alice"}), read_concern=rc_snapshot, write_concern=wc_majority)

updated_alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert updated_alice_balance == new_alice_balance

# add 500 to Bob's account
bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert bob_balance >= amount_to_transfer
new_bob_balance = bob_balance + amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.with_transaction(lambda s: callback(s, new_bob_balance, {"name":
        "Bob"}), read_concern=rc_snapshot, write_concern=wc_majority)

updated_bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert updated_bob_balance == new_bob_balance
Sample Python code with Core api
import pymongo

client = pymongo.MongoClient(<connection_string>)
rc_snapshot = pymongo.read_concern.ReadConcern('snapshot')
wc_majority = pymongo.write_concern.WriteConcern('majority')

# To start, drop and create an account collection and insert balances for both Alice
and Bob
collection = client.get_database("bank").get_collection("account")
collection.drop()
collection.insert_one({"_id": 1, "name": "Alice", "balance": 1000})
collection.insert_one({"_id": 2, "name": "Bob", "balance": 1000})

amount_to_transfer = 500

# deduct 500 from Alice's account
alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert alice_balance >= amount_to_transfer
new_alice_balance = alice_balance - amount_to_transfer
```

```
with client.start_session({'causalConsistency':False}) as session:
    session.start_transaction(read_concern=rc_snapshot, write_concern=wc_majority)
    collection.update_one({"name": "Alice"}, {'$set': {"balance":
new_alice_balance}}, session=session)
    session.commit_transaction()

updated_alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert updated_alice_balance == new_alice_balance

# add 500 to Bob's account
bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert bob_balance >= amount_to_transfer
new_bob_balance = bob_balance + amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.start_transaction(read_concern=rc_snapshot, write_concern=wc_majority)
    collection.update_one({"name": "Bob"}, {'$set': {"balance": new_bob_balance}},
session=session)
    session.commit_transaction()

updated_bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert updated_bob_balance == new_bob_balance
```

核心 API 的交易 API 範例

Javascript

下面的代碼演示如何使用 Amazon DocumentDB 交易 API 與 JavaScript。

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success ***
// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});
```

```
session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert(aliceBalance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"}, {"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert.eq(newAliceBalance, findAliceBalance);

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountColl.update({"name": "Bob"}, {"$set": {"balance": newBobBalance}});
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;
assert.eq(newBobBalance, findBobBalance);

session.commitTransaction();

accountColl.find();
```

C#

下列程式碼示範如何搭配 C# 使用 Amazon DocumentDB 交易 API。

```
// C# Core API

public void TransferMoneyWithRetry(IMongoCollection<BsonDocument> accountColl,
    IClientSessionHandle session)
{
    var amountToTransfer = 500;

    // start transaction
    var transactionOptions = new TransactionOptions(
        readConcern: ReadConcern.Snapshot,
        writeConcern: WriteConcern.WMajority);
    session.StartTransaction(transactionOptions);
    try
    {
        // deduct $500 from Alice's account
        var aliceBalance = accountColl.Find(session,
            Builders<BsonDocument>.Filter.Eq("name",
                "Alice")).FirstOrDefault().GetValue("balance");
```

```
        Debug.Assert(aliceBalance >= amountToTransfer);
        var newAliceBalance = aliceBalance.AsInt32 - amountToTransfer;
        accountColl.UpdateOne(session, Builders<bSondocument>.Filter.Eq("name",
"Alice"),
                                Builders<bSondocument>.Update.Set("balance",
newAliceBalance));
        aliceBalance = accountColl.Find(session,
Builders<bSondocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
        Debug.Assert(aliceBalance == newAliceBalance);

        // add $500 from Bob's account
        var bobBalance = accountColl.Find(session,
Builders<bSondocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
        var newBobBalance = bobBalance.AsInt32 + amountToTransfer;
        accountColl.UpdateOne(session, Builders<bSondocument>.Filter.Eq("name",
"Bob"),
                                Builders<bSondocument>.Update.Set("balance",
newBobBalance));
        bobBalance = accountColl.Find(session,
Builders<bSondocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
        Debug.Assert(bobBalance == newBobBalance);

    }
    catch (Exception e)
    {
        session.AbortTransaction();
        throw;
    }

    session.CommitTransaction();
}

}
public void DoTransactionWithRetry(MongoClient client)
{
    var dbName = "bank";
    var collName = "account";
    using (var session = client.StartSession(new
ClientSessionOptions{CausalConsistency = false}))
    {
        try
```

```
    {
        var bankDB = client.GetDatabase(dbName);
        var accountColl = bankDB.GetCollection<bSondocument>(collName);
        bankDB.DropCollection(collName);
        accountColl.InsertOne(session, new BsonDocument { {"name", "Alice"},
{"balance", 1000 } });
        accountColl.InsertOne(session, new BsonDocument { {"name", "Bob"},
{"balance", 1000 } });

        while(true) {
            try
            {
                TransferMoneyWithRetry(accountColl, session);
                break;
            }
            catch (MongoException e)
            {
                if(e.HasErrorLabel("TransientTransactionError"))
                {
                    continue;
                }
                else
                {
                    throw;
                }
            }
        }

        // check values outside of transaction
        var aliceNewBalance =
accountColl.Find(Builders<bSondocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
        var bobNewBalance =
accountColl.Find(Builders<bSondocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
        Debug.Assert(aliceNewBalance == 500);
        Debug.Assert(bobNewBalance == 1500);
    }
    catch (Exception e)
    {
        Console.WriteLine("Error running transaction: " + e.Message);
    }
}
```



```
}
```

Ruby

下列程式碼示範如何搭配紅寶石使用 Amazon DocumentDB 交易 API。

```
# Ruby Core API

def transfer_money_w_retry(session, accountColl)
  amountToTransfer = 500

  session.start_transaction(read_concern: {level: :snapshot}, write_concern:
  {w: :majority})
  # deduct $500 from Alice's account
  aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
  session).first['balance']
  assert aliceBalance >= amountToTransfer
  newAliceBalance = aliceBalance - amountToTransfer
  accountColl.update_one({"name"=>"Alice"}, { "$set" =>
  {"balance"=>newAliceBalance} }, :session=> session)
  aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
  session).first['balance']
  assert_equal(newAliceBalance, aliceBalance)

  # add $500 to Bob's account
  bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
  session).first['balance']
  newBobBalance = bobBalance + amountToTransfer
  accountColl.update_one({"name"=>"Bob"}, { "$set" =>
  {"balance"=>newBobBalance} }, :session=> session)
  bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
  session).first['balance']
  assert_equal(newBobBalance, bobBalance)

  session.commit_transaction
end

def do_txn_w_retry(client)
  dbName = "bank"
  collName = "account"

  session = client.start_session(:causal_consistency=> false)
  bankDB = Mongo::Database.new(client, dbName)
```

```
accountColl = bankDB[collName]
accountColl.drop()

accountColl.insert_one({"name"=>"Alice", "balance"=>1000})
accountColl.insert_one({"name"=>"Bob", "balance"=>1000})

begin
  transferMoneyWithRetry(session, accountColl)
  puts "transaction committed"
rescue Mongo::Error => e
  if e.label?('TransientTransactionError')
    retry
  else
    puts "transaction failed"
    raise
  end
end

# check results outside of transaction
aliceBalance = accountColl.find({"name"=>"Alice"}).first['balance']
bobBalance = accountColl.find({"name"=>"Bob"}).first['balance']
assert_equal(aliceBalance, 500)
assert_equal(bobBalance, 1500)

end
```

Java

下面的代碼演示如何利用 Amazon DocumentDB 交易 API 與 Java。

```
// Java (sync) - Core API

public void transferMoneyWithRetry() {
  // connect to server
  MongoClientURI mongoURI = new MongoClientURI(uri);
  MongoClient mongoClient = new MongoClient(mongoURI);

  MongoDB database = mongoClient.getDatabase("bank");
  MongoCollection accountColl = database.getCollection("account");
  accountColl.drop();

  // insert some sample data
  accountColl.insertOne(new Document("name", "Alice").append("balance", 1000));
  accountColl.insertOne(new Document("name", "Bob").append("balance", 1000));
}
```

```
while (true) {
    try {
        doTransferMoneyWithRetry(accountColl, mongoClient);
        break;
    } catch (MongoException e) {
        if (e.hasErrorLabel(MongoException.TRANSCIENT_TRANSACTION_ERROR_LABEL)) {
            continue;
        } else {
            throw e;
        }
    }
}

}

public void doTransferMoneyWithRetry(MongoCollection accountColl, MongoClient
mongoClient) {
    int amountToTransfer = 500;

    TransactionOptions txnOptions = TransactionOptions.builder()
        .readConcern(ReadConcern.SNAPSHOT)
        .writeConcern(WriteConcern.MAJORITY)
        .build();
    ClientSessionOptions sessionOptions =
ClientSessionOptions.builder().causallyConsistent(false).build();
    try ( ClientSession clientSession = mongoClient.startSession(sessionOptions) ) {
        clientSession.startTransaction(txnOptions);

        // deduct $500 from Alice's account
        List<Document> documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
"Alice")).into(documentList);
        int aliceBalance = (int) documentList.get(0).get("balance");
        Assert.assertTrue(aliceBalance >= amountToTransfer);
        int newAliceBalance = aliceBalance - amountToTransfer;
        accountColl.updateOne(clientSession, new Document("name", "Alice"), new
Document("$set", new Document("balance", newAliceBalance)));

        // check Alice's new balance
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
"Alice")).into(documentList);
        int updatedBalance = (int) documentList.get(0).get("balance");
        Assert.assertEquals(updatedBalance, newAliceBalance);
    }
}
```

```
        // add $500 to Bob's account
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
"Bob")).into(documentList);
        int bobBalance = (int) documentList.get(0).get("balance");
        int newBobBalance = bobBalance + amountToTransfer;
        accountColl.updateOne(clientSession, new Document("name", "Bob"), new
Document("$set", new Document("balance", newBobBalance)));

        // check Bob's new balance
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
"Bob")).into(documentList);
        updatedBalance = (int) documentList.get(0).get("balance");
        Assert.assertEquals(updatedBalance, newBobBalance);

        // commit transaction
        clientSession.commitTransaction();
    }
}
// Java (async) -- Core API
public void transferMoneyWithRetry() {
    // connect to the server
    MongoClient mongoClient = MongoClient.create(uri);

    MongoDBDatabase bankDB = mongoClient.getDatabase("bank");
    MongoCollection accountColl = bankDB.getCollection("account");
    SubscriberLatchWrapper<Void> dropCallback = new SubscriberLatchWrapper<>();
    mongoClient.getDatabase("bank").drop().subscribe(dropCallback);
    dropCallback.await();

    // insert some sample data
    SubscriberLatchWrapper<InsertOneResult> insertionCallback = new
SubscriberLatchWrapper<>();
    accountColl.insertOne(new Document("name", "Alice").append("balance",
1000)).subscribe(insertionCallback);
    insertionCallback.await();

    insertionCallback = new SubscriberLatchWrapper<>();
    accountColl.insertOne(new Document("name", "Bob").append("balance",
1000)).subscribe(insertionCallback);
    insertionCallback.await();
}
```

```
while (true) {
    try {
        doTransferMoneyWithRetry(accountColl, mongoClient);
        break;
    } catch (MongoException e) {
        if (e.hasErrorLabel(MongoException.TRANSACTION_ERROR_LABEL)) {
            continue;
        } else {
            throw e;
        }
    }
}
```

```
public void doTransferMoneyWithRetry(MongoCollection accountColl, MongoClient
mongoClient) {
    int amountToTransfer = 500;

    // start the transaction
    TransactionOptions txnOptions = TransactionOptions.builder()
        .readConcern(ReadConcern.SNAPSHOT)
        .writeConcern(WriteConcern.MAJORITY)
        .build();

    ClientSessionOptions sessionOptions =
    ClientSessionOptions.builder().causallyConsistent(false).build();

    SubscriberLatchWrapper<ClientSession> sessionCallback = new
SubscriberLatchWrapper<>();
    mongoClient.startSession(sessionOptions).subscribe(sessionCallback);
    ClientSession session = sessionCallback.get().get(0);
    session.startTransaction(txnOptions);

    // deduct $500 from Alice's account
    SubscriberLatchWrapper<Document> findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
"Alice")).first().subscribe(findCallback);
    Document documentFound = findCallback.get().get(0);
    int aliceBalance = (int) documentFound.get("balance");
    int newAliceBalance = aliceBalance - amountToTransfer;

    SubscriberLatchWrapper<UpdateResult> updateCallback = new
SubscriberLatchWrapper<>();
```

```
    accountColl.updateOne(session, new Document("name",
"Alice"), new Document("$set", new Document("balance",
newAliceBalance))).subscribe(updateCallback);
    updateCallback.await();

    // check Alice's new balance
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
"Alice")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    int updatedBalance = (int) documentFound.get("balance");
    Assert.assertEquals(updatedBalance, newAliceBalance);

    // add $500 to Bob's account
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
"Bob")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    int bobBalance = (int) documentFound.get("balance");
    int newBobBalance = bobBalance + amountToTransfer;

    updateCallback = new SubscriberLatchWrapper<>();
    accountColl.updateOne(session, new Document("name", "Bob"), new Document("$set",
new Document("balance", newBobBalance))).subscribe(updateCallback);
    updateCallback.await();

    // check Bob's new balance
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
"Bob")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    updatedBalance = (int) documentFound.get("balance");
    Assert.assertEquals(updatedBalance, newBobBalance);

    // commit the transaction
    SubscriberLatchWrapper<Void> transactionCallback = new
SubscriberLatchWrapper<>();
    session.commitTransaction().subscribe(transactionCallback);
    transactionCallback.await();
}

public class SubscriberLatchWrapper<T> implements Subscriber<T> {

    /**
```

```
    * A Subscriber that stores the publishers results and provides a latch so can
    block on completion.
    *
    * @param <T> The publishers result type
    */
    private final List<T> received;
    private final List<RuntimeException> errors;
    private final CountdownLatch latch;
    private volatile Subscription subscription;
    private volatile boolean completed;

    /**
     * Construct an instance
     */
    public SubscriberLatchWrapper() {
        this.received = new ArrayList<>();
        this.errors = new ArrayList<>();
        this.latch = new CountdownLatch(1);
    }

    @Override
    public void onSubscribe(final Subscription s) {
        subscription = s;
        subscription.request(Integer.MAX_VALUE);
    }

    @Override
    public void onNext(final T t) {
        received.add(t);
    }

    @Override
    public void onError(final Throwable t) {
        if (t instanceof RuntimeException) {
            errors.add((RuntimeException) t);
        } else {
            errors.add(new RuntimeException("Unexpected exception", t));
        }
        onComplete();
    }

    @Override
    public void onComplete() {
        completed = true;
    }
}
```

```
        subscription.cancel();
        latch.countDown();
    }

    /**
     * Get received elements
     *
     * @return the list of received elements
     */
    public List<T> getReceived() {
        return received;
    }

    /**
     * Get received elements.
     *
     * @return the list of receive elements
     */
    public List<T> get() {
        return await().getReceived();
    }

    /**
     * Await completion or error
     *
     * @return this
     */
    public SubscriberLatchWrapper<T> await() {
        subscription.request(Integer.MAX_VALUE);
        try {
            if (!latch.await(300, TimeUnit.SECONDS)) {
                throw new MongoTimeoutException("Publisher onComplete timed out for
300 seconds");
            }
        } catch (InterruptedException e) {
            throw new MongoInterruptedException("Interrupted waiting for
observation", e);
        }
        if (!errors.isEmpty()) {
            throw errors.get(0);
        }
        return this;
    }
}
```



```
public boolean getCompleted() {
    return this.completed;
}

public void close() {
    subscription.cancel();
    received.clear();
}
}
```

C

下列程式碼示範如何搭配 C 使用 Amazon DocumentDB 交易 API。

```
// Sample C code with core session

bool core_session(mongoc_client_session_t *client_session, mongoc_collection_t*
collection, bson_t *selector, int64_t balance){
    bool r = true;
    bson_error_t error;
    bson_t *opts = bson_new();
    bson_t *update = BCON_NEW ("$set", "{", "balance", BCON_INT64 (balance), "}");

    // set read & write concern
    mongoc_read_concern_t *read_concern = mongoc_read_concern_new ();
    mongoc_write_concern_t *write_concern = mongoc_write_concern_new ();
    mongoc_transaction_opts_t *txn_opts = mongoc_transaction_opts_new ();

    mongoc_write_concern_set_w(write_concern, MONGOC_WRITE_CONCERN_W_MAJORITY);
    mongoc_read_concern_set_level(read_concern, MONGOC_READ_CONCERN_LEVEL_SNAPSHOT);
    mongoc_transaction_opts_set_write_concern (txn_opts, write_concern);
    mongoc_transaction_opts_set_read_concern (txn_opts, read_concern);

    mongoc_client_session_start_transaction (client_session, txn_opts, &error);
    mongoc_client_session_append (client_session, opts, &error);

    r = mongoc_collection_update_one (collection, selector, update, opts, NULL,
&error);

    mongoc_client_session_commit_transaction (client_session, NULL, &error);
    bson_destroy (opts);
    mongoc_transaction_opts_destroy(txn_opts);
    mongoc_read_concern_destroy(read_concern);
}
```

```
    mongoc_write_concern_destroy(write_concern);
    bson_destroy (update);
    return r;
}

void test_core_money_transfer(mongoc_client_t* client, mongoc_collection_t*
collection, int amount_to_transfer){

    bson_t reply;
    bool r = true;
    const bson_t *doc;
    bson_iter_t iter;
    bson_error_t error;

    // find query
    bson_t *alice_query = bson_new ();
    BSON_APPEND_UTF8(alice_query, "name", "Alice");

    bson_t *bob_query = bson_new ();
    BSON_APPEND_UTF8(bob_query, "name", "Bob");

    // create session
    // set causal consistency to false
    mongoc_session_opt_t *session_opts = mongoc_session_opts_new ();
    mongoc_session_opts_set_causal_consistency (session_opts, false);
    // start the session
    mongoc_client_session_t *client_session = mongoc_client_start_session (client,
session_opts, &error);

    // add session to options
    bson_t *opts = bson_new();
    mongoc_client_session_append (client_session, opts, &error);

    // deduct 500 from Alice
    // find account balance of Alice
    mongoc_cursor_t *cursor = mongoc_collection_find_with_opts (collection,
alice_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t alice_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(alice_balance >= amount_to_transfer);
    int64_t new_alice_balance = alice_balance - amount_to_transfer;
```

```
// core
r = core_session (client_session, collection, alice_query, new_alice_balance);
assert(r);

// find account balance of Alice after transaction
cursor = mongoc_collection_find_with_opts (collection, alice_query, NULL, NULL);
mongoc_cursor_next (cursor, &doc);
bson_iter_init (&iter, doc);
bson_iter_find (&iter, "balance");
alice_balance = (bson_iter_value (&iter))->value.v_int64;
assert(alice_balance == new_alice_balance);
assert(alice_balance == 500);

// add 500 to Bob's balance
// find account balance of Bob
cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
mongoc_cursor_next (cursor, &doc);
bson_iter_init (&iter, doc);
bson_iter_find (&iter, "balance");
int64_t bob_balance = (bson_iter_value (&iter))->value.v_int64;
int64_t new_bob_balance = bob_balance + amount_to_transfer;

//core
r = core_session (client_session, collection, bob_query, new_bob_balance);
assert(r);

// find account balance of Bob after transaction
cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
mongoc_cursor_next (cursor, &doc);
bson_iter_init (&iter, doc);
bson_iter_find (&iter, "balance");
bob_balance = (bson_iter_value (&iter))->value.v_int64;
assert(bob_balance == new_bob_balance);
assert(bob_balance == 1500);

// cleanup
bson_destroy(alice_query);
bson_destroy(bob_query);
mongoc_client_session_destroy(client_session);
bson_destroy(opts);
mongoc_cursor_destroy(cursor);
bson_destroy(doc);
}
```

```
int main(int argc, char* argv[]) {
    mongoc_init ();
    mongoc_client_t* client = mongoc_client_new (<connection uri>);
    bson_error_t error;

    // connect to bank db
    mongoc_database_t *database = mongoc_client_get_database (client, "bank");
    // access account collection
    mongoc_collection_t* collection = mongoc_client_get_collection(client, "bank",
"account");
    // set amount to transfer
    int64_t amount_to_transfer = 500;
    // delete the collection if already existing
    mongoc_collection_drop(collection, &error);

    // open Alice account
    bson_t *alice_account = bson_new ();
    BSON_APPEND_UTF8(alice_account, "name", "Alice");
    BSON_APPEND_INT64(alice_account, "balance", 1000);

    // open Bob account
    bson_t *bob_account = bson_new ();
    BSON_APPEND_UTF8(bob_account, "name", "Bob");
    BSON_APPEND_INT64(bob_account, "balance", 1000);

    bool r = true;

    r = mongoc_collection_insert_one(collection, alice_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}
    r = mongoc_collection_insert_one(collection, bob_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}

    test_core_money_transfer(client, collection, amount_to_transfer);

}
```

Scala

下面的代碼演示了如何利用 Amazon DocumentDB 交易 API 與斯卡拉。

```
// Scala Core API
def transferMoneyWithRetry(sessionObservable: SingleObservable[ClientSession] ,
    database: MongoDatabase ): Unit = {
    val accountColl = database.getCollection("account")
```

```
var amountToTransfer = 500

var transactionObservable: Observable[ClientSession] =
sessionObservable.map(clientSession => {
  clientSession.startTransaction()

  // deduct $500 from Alice's account
  var aliceBalance = accountColl.find(clientSession, Document("name" ->
"Alice")).await().head.getInteger("balance")
  assert(aliceBalance >= amountToTransfer)
  var newAliceBalance = aliceBalance - amountToTransfer
  accountColl.updateOne(clientSession, Document("name" -> "Alice"),
Document("$set" -> Document("balance" -> newAliceBalance))).await()
  aliceBalance = accountColl.find(clientSession, Document("name" ->
"Alice")).await().head.getInteger("balance")
  assert(aliceBalance == newAliceBalance)

  // add $500 to Bob's account
  var bobBalance = accountColl.find(clientSession, Document("name" ->
"Bob")).await().head.getInteger("balance")
  var newBobBalance = bobBalance + amountToTransfer
  accountColl.updateOne(clientSession, Document("name" -> "Bob"), Document("$set"
-> Document("balance" -> newBobBalance))).await()
  bobBalance = accountColl.find(clientSession, Document("name" ->
"Bob")).await().head.getInteger("balance")
  assert(bobBalance == newBobBalance)

  clientSession
})

transactionObservable.flatMap(clientSession =>
clientSession.commitTransaction()).await()
}

def doTransactionWithRetry(): Unit = {
  val client: MongoClient = MongoClientWrapper.getMongoClient()
  val database: MongoDBDatabase = client.getDatabase("bank")
  val accountColl = database.getCollection("account")
  accountColl.drop().await()

  val sessionOptions =
ClientSessionOptions.builder().causallyConsistent(false).build()
  var sessionObservable: SingleObservable[ClientSession] =
client.startSession(sessionOptions)
```

```
accountColl.insertOne(Document("name" -> "Alice", "balance" -> 1000)).await()
accountColl.insertOne(Document("name" -> "Bob", "balance" -> 1000)).await()

var retry = true
while (retry) {
    try {
        transferMoneyWithRetry(sessionObservable, database)
        println("transaction committed")
        retry = false
    }
    catch {
        case e: MongoException if
e.hasErrorLabel(MongoException.TRANSIENT_TRANSACTION_ERROR_LABEL) => {
            println("retrying transaction")
        }
        case other: Throwable => {
            println("transaction failed")
            retry = false
            throw other
        }
    }
}

// check results outside of transaction
assert(accountColl.find(Document("name" ->
"Alice")).results().head.getInteger("balance") == 500)
assert(accountColl.find(Document("name" ->
"Bob")).results().head.getInteger("balance") == 1500)

accountColl.drop().await()
}
```

支援的 命令

| 命令 | 支援 |
|-------------------|----|
| abortTransaction | 是 |
| commitTransaction | 是 |

| 命令 | 支援 |
|---------------------------------------|----|
| <code>endSessions</code> | 是 |
| <code>killSession</code> | 是 |
| <code>killAllSession</code> | 是 |
| <code>killAllSessionsByPattern</code> | 否 |
| <code>refreshSessions</code> | 否 |
| <code>startSession</code> | 是 |

不支援

| 方法 | 階段或指令 |
|---|--|
| <code>db.collection.aggregate()</code> | <code>\$collStats</code> <code>\$currentOp</code> <code>\$indexStats</code> <code>\$listSessions</code> <code>\$out</code> |
| <code>db.collection.count()</code> | <code>\$where</code> |
| <code>db.collection.countDocuments()</code> | <code>\$near</code> <code>\$nearSphere</code> |
| <code>db.collection.insert()</code> | <p><code>insert</code> 如果未針對現有集合執行，則不受支援。如果它以預先存在的集合為目標，則支援此方法。</p> |

工作階段

MongoDB 會話是用於支持可重試寫入，因果一致性，事務和跨數據庫管理操作的框架。建立工作階段時，用戶端會產生邏輯工作階段識別碼 (lsid)，並在傳送命令至伺服器時，用來標記該工作階段內的所有作業。

Amazon DocumentDB 支援使用工作階段來啟用交易，但不支援因果一致性或可重試寫入。

使用 Amazon DocumentDB 中的交易時，將使用 `session.startTransaction()` API 從工作階段內啟動交易，而工作階段一次支援單一交易。同樣地，交易也使用提交 (`session.commitTransaction()`) 或中止 (`session.abortTransaction()`) API 來完成。

因果一致性

因果一致性保證在單個客戶端會話中，客戶端將觀察 read-after-write 一致性，單原子讀取/寫入以及寫入將遵循讀取，並且這些保證適用於集群中的所有實例，而不僅僅是主實例。Amazon DocumentDB 不支持因果一致性，下面的語句將導致錯誤。

```
var mySession = db.getMongo().startSession();
var mySessionObject = mySession.getDatabase('test').getCollection('account');

mySessionObject.updateOne({"_id": 2}, {"$inc": {"balance": 400}});
//Result:{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

mySessionObject.find()
//Error: error: {
//      "ok" : 0,
//      "code" : 303,
//      "errmsg" : "Feature not supported: 'causal consistency'",
//      "operationTime" : Timestamp(1603461817, 493214)
//}

mySession.endSession()
```

您可以在工作階段中停用因果一致性。請注意，這樣做將使您能夠利用會話框架，但不會為讀取提供因果一致性保證。使用 Amazon DocumentDB 時，主要執行個體的讀取將 read-after-write 保持一致，並且從複本執行個體讀取最終保持一致。交易是利用工作階段的主要使用案例。

```
var mySession = db.getMongo().startSession({causalConsistency: false});
```



```
var mySessionObject = mySession.getDatabase('test').getCollection('account');

mySessionObject.updateOne({"_id": 2}, {"$inc": {"balance": 400}});
//Result:{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

mySessionObject.find()
//{ "_id" : 1, "name" : "Bob", "balance" : 100 }
//{ "_id" : 2, "name" : "Alice", "balance" : 1700 }
```

可重試寫入

可重試寫入是一項功能，當發生網路錯誤或用戶端找不到主要作業時，用戶端會嘗試一次重試寫入作業。在 Amazon DocumentDB 中，不支援可重試寫入，而且必須停用。您可以使用連接字符串中的命令 (`retryWrites=false`) 禁用它。

例外：如果您使用的是 mongo shell，請不要在任何代碼字符串中包含該 `retryWrites=false` 命令。根據預設，可重試寫入都會停用。包括 `retryWrites=false` 可能會導致正常讀取命令失敗。

交易錯誤

當使用事務，有些情況下，可以 `yield` 一個錯誤，指出一個事務編號不匹配任何正在進行的事務。

至少可以在兩種不同的情況下產生錯誤：

- After the one-minute transaction timeout.
- After an instance restart (due to patching, crash recovery, etc.), it is possible to receive this error even in cases where the transaction successfully committed. During an instance restart, the database can't tell the difference between a transaction that successfully completed versus a transaction that aborted. In other words, the transaction completion state is ambiguous.

處理此錯誤的最佳方法是使事務更新冪等-例如，通過使用 `$set` mutator 而不是遞增/遞減操作。見下文：

```
{ "ok" : 0,
  "operationTime" : Timestamp(1603938167, 1),
  "code" : 251,
  "errmsg" : "Given transaction number 1 does not match any in-progress transactions."
}
```

Amazon DocumentDB 的最佳實踐

了解使用 Amazon DocumentDB 的最佳實務 (與 MongoDB 相容性)。在新最佳實務確定時，會不斷更新此小節。

主題

- [基本操作準則](#)
- [執行個體大小設定](#)
- [使用索引](#)
- [安全最佳實務](#)
- [成本最佳化](#)
- [使用指標來識別效能問題](#)
- [TTL 和時間序列工作負載](#)
- [遷移](#)
- [使用叢集參數群組](#)
- [彙總管道查詢](#)
- [batchInsert 和 batchUpdate](#)

基本操作準則

以下是每個人在使用 Amazon DocumentDB 時都應遵循的基本操作準則。Amazon DocumentDB 服務等級協議要求您遵循下列準則。

- 在兩個 AWS 可用區域中部署由兩個或多個 Amazon DocumentDB 執行個體組成的叢集。對於生產工作負載，我們建議在三個可用區域中部署由三個或更多 Amazon DocumentDB 執行個體組成的叢集。
- 在指定的服務限制內使用服務。如需詳細資訊，請參閱 [Amazon DocumentDB 配額和限制](#)。
- 監控您的記憶體、CPU、連線和儲存體用量。為了協助您維持系統效能和可用性，請設 CloudWatch 定 Amazon 以在使用模式變更或接近部署容量時通知您。
- 當您接近容量限制時擴展您的執行個體。您的執行個體應該佈建足夠的運算資源 (例如 RAM、CPU)，以因應應用程式不可預見的需求增加。
- 設定備份保留期以符合您的復原點目標。

- 測試叢集的容錯移轉，以了解您的使用案例執行此程序需時多長。如需詳細資訊，請參閱 [Amazon DocumentDB 容錯移轉](#)。
- 使用叢集端點 (請參閱[Amazon DocumentDB 端點](#)) 和複本集模式 (請參閱) Connect 到 Amazon DocumentDB 叢集，將容錯移轉對應用程式的影響降到最低。以複本集形式連接至 [Amazon DocumentDB](#)
- 選擇驅動程式讀取偏好設定，以發揮最大讀取擴展，同時符合您應用程式的讀取一致性要求。secondaryPreferred 讀取偏好設定會啟用複本讀取，並釋出主要執行個體以進行更多工作。如需詳細資訊，請參閱 [讀取偏好設定選項](#)。
- 設計您的應用程式，以便在網路和資料庫發生錯誤時，保有故障恢復的應變能力。使用您驅動程式的錯誤機制，區分暫時性錯誤或永久性錯誤。適當時，使用指數退避機制重新嘗試暫時性錯誤。確認您的應用程式在實作重試邏輯時會考慮資料一致性。
- 為所有生產叢集或任何具有重要資料的叢集啟用叢集刪除保護。在刪除 Amazon DocumentDB 叢集之前，請先建立最終快照。如果您使用部署資源 AWS CloudFormation，請啟用終止保護。如需詳細資訊，請參閱 [終止保護及刪除保護](#)。
- 建立 Amazon DocumentDB 叢集時，引擎版本是選用的參數，預設為最新的主要引擎版本。目前的主要引擎版本為 4.0.0。當新的主要引擎版本發行時，--engine 版本的預設引擎版本將會更新，以反映最新的主要引擎版本。因此，對於生產工作負載，尤其是依賴指令碼、自動化或 AWS CloudFormation 範本的工作負載，我們建議您將--engine-version 明確指定為預期的主要版本。

執行個體大小設定

在 Amazon DocumentDB 中選擇執行個體大小最關鍵的方面之一就是快取的 RAM 容量。Amazon DocumentDB 會為自己的服務保留三分之一的記憶體，這表示只有三分之二的執行個體 RAM 可供快取使用。因此，選擇具有足夠 RAM 的執行個體類型以適合記憶體中的工作集 (亦即資料和索引) 是 Amazon DocumentDB 的最佳實務。具有適當大小的執行個體將有助於最佳化整體效能，並可能將 I/O 成本降至最低。您可以使用第三方 [Amazon DocumentDB 大小計算器來估算](#) 特定工作負載的執行個體大小。

若要判斷應用程式的工作集是否適合記憶體，請 CloudWatch 針對負載下的叢集中每個執行個體監控 BufferCacheHitRatio 使用 Amazon 的情況。

此指 BufferCacheHitRatio CloudWatch 標會測量從執行個體記憶體快取 (相對於儲存體磁碟區) 提供的資料和索引百分比。一般來說，BufferCacheHitRatio 的數值應該盡可能高，因為從工作集記憶體讀取資料比從儲存磁碟區讀取更快速且更具成本效益。儘管盡可能保持 BufferCacheHitRatio 接近 100% 是理想的做法，但是最可能達到的數值將取決於您的應用程式的存取模式和效能需求。為

了盡可能保持最高的 BufferCacheHitRatio，建議您叢集中的執行個體佈建足夠的 RAM，以便在記憶體中容納索引和工作資料集。

如果您的索引不適用記憶體，您會看到一個較低的 BufferCacheHitRatio。持續從磁碟讀取會產生額外的 I/O 成本，而且效能不如從記憶體讀取來得好。如果您的 BufferCacheHitRatio 比例低於預期，請為您的叢集擴充執行個體大小，以提供更多 RAM 以符合記憶體中的工作集資料。如果擴大執行個體類別會導致 BufferCacheHitRatio 大幅增加，則應用程式的工作集不適用記憶體。持續擴展直到擴展操作後 BufferCacheHitRatio 不再大幅增加。如需監控執行個體指標的相關資訊，請參閱 [Amazon DocumentDB 指標](#)。

根據您的工作負載和延遲需求，您的應用程式在穩定狀態使用期間具有較高的 BufferCacheHitRatio 值，而在執行個體上執行需要掃描整個集合的分析查詢時，BufferCacheHitRatio 時而下降，是可接受的情況。BufferCacheHitRatio 中這些定期的下降對後續的查詢可能會顯示為較高的延遲，因為需要將工作集資料從儲存磁碟區重新填入緩衝區快取。我們建議您先在生產前環境中使用典型的生產工作負載測試工作負載，以便在將工作負載部署至生產環境之前了解效能特性和 **BufferCacheHitRatio**。

BufferCacheHitRatio 是執行個體特定的指標，因此相同叢集中的不同執行個體可能會有不同的 BufferCacheHitRatio 值，視讀取作業在主要和複本執行個體之間的分配情況而定。如果您的操作工作負載無法處理在執行分析查詢後重新植入工作集快取時而增加的延遲，您應該嘗試將一般工作負載的緩衝區快取與分析查詢隔離。您可以將操作查詢導向至主要執行個體，而只將分析查詢導向至複本執行個體，以達到完全 BufferCacheHitRatio 隔離。您也可以透過將分析查詢導向至特定複本執行個體，在了解某些百分比的一般查詢也會在該複本上執行，且可能受到影響的情況下，達成部分隔離。

適當的 BufferCacheHitRatio 值取決於您的使用案例和應用程式需求。此指標沒有最佳或最小值；只有您自己可以決定從成本和效能的角度來看，是否可以接受暫時較低 BufferCacheHitRatio 的折衷。

使用索引

建立索引

將資料匯入 Amazon DocumentDB 時，您應該先建立索引，然後再匯入大型資料集。您可以使用 [Amazon 文件資料庫索引工具從執行中的 MongoDB 執行個體或mongodump目錄擷取索引](#)，並在 [Amazon DocumentDB](#) 叢集中建立這些索引。如需遷移的詳細指導方針，請參閱 [遷移到 Amazon DocumentDB](#)。

索引選擇性

我們建議您限制建立索引的欄位，其中重複值的數目會小於集合中文件總數的 1%。例如，如果您的集合包含 100,000 份文件，則只在相同值出現 1000 次或更少的欄位上建立索引。

選擇具有大量唯一值 (即高基數) 的索引可確保篩選器作業傳回少量文件，從而在索引掃描期間產生良好的效能。高基數索引範例像是唯一索引，它可以保證相等述詞最多只能傳回單一文件。低基數索引的範例像是布林欄位，以及對一週中某天的索引。由於效能不佳，資料庫的查詢最佳化工具不太可能選擇低基數索引。同時，低基數索引會繼續消耗磁碟空間和 I/O 等資源。根據經驗法則，您應該在典型值頻率為總集合大小的 1% 或更小的字段上鎖定索引。

此外，建議只在通常用作篩選器的欄位上建立索引，並定期尋找未使用的索引。如需詳細資訊，請參閱 [如何分析索引使用情況並識別未使用的索引？](#)。

索引對編寫數據的影響

雖然索引可以透過避免掃描集合中的每個文件來改善查詢性能，但此改進方式有所權衡。對於集合上的每個索引，每次插入、更新或刪除文件時，資料庫必須更新集合並將欄位寫入集合的每個索引。例如，如果集合有九個索引，資料庫必須先執行十次寫入，才能確認用戶端的作業。因此，每個額外的索引都會造成額外的寫入延遲、I/O 以及整體使用的儲存量增加。

叢集執行個體的大小必須適當，才能保留所有工作集記憶體。如此可避免從儲存磁碟區持續讀取索引頁面的需求，進而對效能造成負面影響，並產生較高的 I/O 成本。如需詳細資訊，請參閱 [執行個體大小設定](#)。

為了獲得最佳效能，請盡量減少集合中的索引數目，只新增必要的索引，以改善常見查詢的效能。雖然工作負載有所不同，但好的指導方針是將每個集合的索引數目保持在五個或更少。

識別遺失索引

識別缺少的索引是我們建議定期執行的最佳做法。如需詳細資訊，請參閱 [如何識別遺失的索引？](#)。

識別未使用的索引

識別和移除未使用的索引是我們建議定期執行的最佳實務。如需詳細資訊，請參閱 [如何分析索引使用情況並識別未使用的索引？](#)。

安全最佳實務

為了獲得安全最佳實務，您必須使用 AWS Identity and Access Management (IAM) 帳戶來控制對 Amazon DocumentDB API 操作的存取，尤其是建立、修改或刪除 Amazon DocumentDB 資源的操作。這類資源包含叢集、安全群組和參數群組。您還必須使用 IAM 來控制執行常見管理動作的動作，例如備份還原叢集。建立 IAM 角色時，請採用最低權限原則。

- 使用[角色類型存取控制](#)強制執行最低權限。
- 為管理 Amazon 文件資源的每個人指派個別身分與存取權管理帳戶。請勿使用 AWS 帳戶 根使用者來管理 Amazon DocumentDB 源。為每個人 (包含您) 建立 IAM 使用者。
- 授予每個 IAM 使用者執行職責所需的最低權限集。
- 使用 IAM 群組來有效管理多個使用者的許可。如需 IAM 的詳細資訊，請參閱[IAM 使用者指南](#)。如需 IAM 最佳實務的相關資訊，請參閱[IAM 最佳實務](#)。
- 定期輪替您的 IAM 登入資料。
- 將 AWS Secrets Manager 設定為自動輪替 Amazon DocumentDB 的密碼。如需詳細資訊，請參閱[AWS 秘密管理員](#)使用指南中的[旋轉您的秘密 Secrets Manager 和 Amazon DocumentDB 的旋轉機AWS 密](#)。
- 授予每個 Amazon DocumentDB 使用者執行其職責所需的最低許可集。如需詳細資訊，請參閱[使用角色型存取控制存取資料庫](#)。
- 使用傳輸層安全性 (TLS) 來加密傳輸中的資料，並 AWS KMS 加密靜態資料。

成本最佳化

下列最佳實務可協助您在使用 Amazon DocumentDB 時管理並將成本降至最低。如需定價資訊，請參閱[Amazon DocumentDB \(與 MongoDB 相容性\) 定價](#)和[Amazon DocumentDB \(與 MongoDB 相容性\) 常見問答集](#)。

- 建立帳單提醒，閾值為您預期的當月帳單的 50% 和 75%。如需有關建立帳單提醒的詳細資訊，請參閱[建立帳單警示](#)。
- Amazon DocumentDB 的架構可將儲存和運算區隔開來，因此即使是單一執行個體叢集也非常耐用。叢集儲存磁碟區在三個可用區域之間以六個方向複寫資料，無論叢集的執行個體數量為何，都能提供極高的耐用性。典型的生產叢集可以包含三種或多種執行個體，以提供高可用性。不過，如果不需要高可用性，則您可以使用單一執行個體開發叢集來使成本達到最佳化。
- 對於開發和測試案例，請停止不再需要的叢集，並且在恢復開發時啟動叢集。如需詳細資訊，請參閱[停止和啟動 Amazon DocumentDB 集群](#)。

- TTL 和變更串流在資料寫入、讀取和刪除時都會產生 I/O。如果您已啟用這些功能，但未在應用程式中使用這些功能，停用這些功能有助於降低成本。

使用指標來識別效能問題

若要識別資源不足和其他常見瓶頸造成的效能問題，您可以監控 Amazon DocumentDB 叢集的可用指標。

檢視效能指標

定期監控效能指標以查看各種時間範圍的平均值、最大值和最小值。效能降級時您便可以得知。您也可以為特定的測量結果閾值設定 Amazon CloudWatch 警示，以便在達到警示時收到警示。

為了對效能問題進行故障診斷，請務必了解系統的基礎效能。您設定新的叢集並執行一般工作負載後，請針對不同間隔 (例如 1 小時、24 小時、1 週、2 週) 的所有效能指標擷取平均、最大值和最低值。這可讓您了解正常運作情況為何。這有助於比較尖峰與離峰時段的操作。之後您可以使用此資訊來得知效能是否下降至標準層級之下。

您可以使用 AWS Management Console 或來檢視效能測量結果 AWS CLI。如需詳細資訊，請參閱 [檢視 CloudWatch 資料](#)。

設定 CloudWatch 鬧鐘

要設置 CloudWatch 警報，請參閱 [Amazon 用 CloudWatch 戶指南中的使用 Amazon CloudWatch 警報](#)。

評估效能指標

執行個體含有幾種不同的指標類別。您判定值是否為可接受，取決於指標。

CPU

- CPU 使用率 — 使用的電腦處理能力百分比。

記憶體

- 可用記憶體 — 執行個體上有多少 RAM 可用。
- 交換使用量 — 執行個體使用了多少交換空間，以 MB 為單位。

輸入/輸出操作

- 讀取 IOPS、寫入 IOPS — 每秒的平均磁碟讀取或寫入作業數。
- 讀取延遲、寫入延遲 — 讀取或寫入作業的平均時間 (毫秒)。
- 讀取輸送量、寫入輸送量 — 每秒從磁碟讀取或寫入磁碟的平均 MB 數。
- 磁碟佇列深度 — 等待寫入磁碟或從磁碟讀取的 I/O 作業數目。

網路流量

- 網路接收輸送量、網路傳輸輸送量 — 進出執行個體的網路流量速率 (以每秒 MB 為單位)。

資料庫連線

- 資料庫連線 — 連線至執行個體的用戶端工作階段數目。

一般來說，效能指標的可接受值視您的基準為何，以及您的應用程式所做之事而定。調查距離基準的一致或趨勢變異。

以下是關於特定類型指標的建議：

- 高 CPU 消耗量 — 高 CPU 消耗量值可能是適當的，前提是它們符合您應用程式的目標 (例如輸送量或並行)，並且是預期的。如果您的 CPU 消耗量持續超過 80%，請考慮擴展您的執行個體。
- 高 RAM 消耗量 — 如果您的FreeableMemory指標經常降到執行個體記憶體總量的 10% 以下，請考慮擴展執行個體。如需 DocumentDB 執行個體遇到高記憶體壓力時會發生什麼情況的詳細資訊，請參閱 [Amazon Document DB 資源控管](#)。
- 交換使用量 — 此量度應保持在零或接近零。如果您的交換用量相當龐大，請考慮擴展您的執行個體。
- 網路流量 — 對於網路流量，請洽詢您的系統管理員，瞭解網域網路和網際網路連線的預期輸送量。調查網路流量的傳輸量是否如預期一致地降低。
- 資料庫連線 — 如果您看到大量的使用者連線，而且執行個體效能和回應時間降低，請考慮限制資料庫連線。執行個體使用者連接的最佳數量，將因執行個體類別和要執行的操作複雜性而不同。如有任何效能指標的問題，您可以執行以改善效能的其中一個動作是調校最常使用與最昂貴的查詢，以了解該調整是否降低對系統資源的壓力。

如果您的查詢經過調整且問題仍然存在，請考慮將 Amazon DocumentDB 執行個體類別升級為具有更多資源 (CPU、RAM、磁碟空間、網路頻寬、I/O 容量) 的執行個體類別。

調校查詢

改善叢集執行個體效能的一個最佳方式是調校最常使用與資源最密集的查詢，讓執行成本較不昂貴。

您可以使用 Profiler (請參閱 [分析亞馬遜文檔數據庫操作](#)) 來記錄叢集上執行之操作的執行時間和詳細資訊。Profiler 適用於監控叢集上最慢的操作，以協助您改善個別查詢效能和整體叢集效能。

您也可以使用 explain 命令了解如何分析特定查詢的查詢計劃。使用此資訊可修改查詢或基礎集合，藉此改善查詢效能 (例如，新增索引)。

TTL 和時間序列工作負載

TTL 指數到期後刪除文件是最勞心費神的過程。本公司不保證在任何特定期間內刪除文件。執行個體大小、執行個體資源使用率、文件大小、整體輸送量、索引數目以及索引和工作集是否適用記憶體等因素，都會影響 TTL 處理程序刪除過期文件的時間。

當 TTL 監控器刪除文件時，每個刪除都會產生 IO 成本，使帳單金額增加。如果輸送量和 TTL 刪除率提高，您可預期會因為 I/O 使用量增加而產生較高的費用。但是，如果您不建立 TTL 索引來刪除文件，而是根據時間將文件分割到集合中，而是在不再需要時將這些集合刪除，則不會產生任何 IO 成本。這比使用 TTL 索引更具成本效益。

對於時間序列工作負載，您可以考慮建立滾動集合而非 TTL 索引，因為滾動集合可以是更有效率的方式來刪除資料，減少密集使用 I/O 的情況。如果您有大型集合 (特別是超過 1TB 的集合) 或 TTL 刪除 I/O 成本的考量，建議您依時間將文件分割為集合，並在不再需要文件時刪除集合。您可以資料擷取率為根據，每天或每個禮拜建立一個集合。雖然需求將根據您的應用程式而有所不同，但好的經驗法是擁有更小的集合而不是一些大型集合。刪除這些集合不會產生 IO 成本，且成本效益明顯比使用 TTL 索引還高。

遷移

最佳實務是，我們建議您在將資料遷移到 Amazon DocumentDB 時，先在 Amazon DocumentDB 中建立索引，然後再遷移資料。首先建立索引可以縮短整體時間並提高遷移速度。要做到這一點，您可以使用 Amazon DocumentDB [索引](#) 工具。如需有關移轉的詳細資訊，請參閱 [Amazon DocumentDB 移轉指南](#)。

我們還建議您在遷移生產資料庫之前，考慮到功能、效能、操作和成本，在 Amazon DocumentDB 上完整測試應用程式是最佳實務。

使用叢集參數群組

建議您在測試叢集執行個體上嘗試進行叢集參數群組變更，再將參數群組變更套用至生產叢集執行個體。如需備份叢集的詳細資訊，請參閱在 [Amazon DocumentDB 中備份和還原](#)。

彙總管道查詢

建立具有多個階段的彙總管道查詢，並僅評估查詢中的一部分資料時，請使用 `$match` 階段做為第一個階段或管道的開頭。使用 `$match` 會先減少彙總管道查詢中需要處理的文件後續階段數量，從而提高查詢的效能。

`batchInsert` 和 `batchUpdate`

執行高速並行和 (`batchInsert`或) `batchUpdate` 作業，且主要執行處理上的 `FreeableMemory` (`CloudWatch` 指標) 數量變為零時，您可以減少批次插入的並行性，或者如果無法減少並行工作負載，請增加執行個體大小以增加的數量。`FreeableMemory`

功能差異：Amazon DocumentDB 和 MongoDB

以下是 Amazon DocumentDB (與 MongoDB 兼容性) 和 MongoDB 之間的功能差異。

主題

- [Amazon DocumentDB 的功能優勢](#)
- [更新的功能差異](#)
- [MongoDB 的功能差異](#)

Amazon DocumentDB 的功能優勢

隱含交易

在 Amazon DocumentDB 中，所有 CRUD 陳述式 (、findAndModify、delete) 都能保證原子性和一致性 updateinsert，即使是修改多個文件的作業也是如此。隨著 Amazon DocumentDB 4.0 的推出，現在支援為多重陳述式和多收集操作提供 ACID 屬性的明確交易。有關在 Amazon DocumentDB 中使用交易的更多信息，請參閱 [交易](#)

以下是 Amazon DocumentDB 中的操作範例，這些操作會修改多個滿足原子和一致性行為的文件。

```
db.miles.update(  
  { "credit_card": { $eq: true } },  
  { $mul: { "flight_miles.$[]": NumberInt(2) } },  
  { multi: true }  
)
```

```
db.miles.updateMany(  
  { "credit_card": { $eq: true } },  
  { $mul: { "flight_miles.$[]": NumberInt(2) } }  
)
```

```
db.runCommand({  
  update: "miles",  
  updates: [  
    {
```

```
    q: { "credit_card": { $eq: true } },
    u: { $mul: { "flight_miles.$[]": NumberInt(2) } },
    multi: true
  }
]
}))
```

```
db.products.deleteMany({
  "cost": { $gt: 30.00 }
})
```

```
db.runCommand({
  delete: "products",
  deletes: [{ q: { "cost": { $gt: 30.00 } }, limit: 0 }]
})
```

構成 `updateMany` 和 `deleteMany` 之類批量操作的個別操作是不可部分完成，但整個批量操作是可部分完成。例如，如果個別插入操作成功執行而沒有錯誤，則整個 `insertMany` 操作都是不可部分完成。如果 `insertMany` 操作遇到錯誤，則 `insertMany` 操作內的每個單獨插入陳述式都將當作不可部分完成操作來執行。如果您需要 `insertMany`、`updateMany`、和 `deleteMany` 作業的 ACID 屬性，建議您使用交易。

更新的功能差異

Amazon DocumentDB 透過客戶要求我們建置的功能向後運作，持續改善與 MongoDB 的相容性。本節包含我們在 Amazon DocumentDB 中移除的功能差異，這些差異可讓客戶更輕鬆地進行遷移和建置應用程式。

主題

- [編製陣列索引](#)
- [多金鑰索引](#)
- [字串中的 Null 字元](#)
- [以角色為基礎的存取控制](#)
- [\\$regex 索引功能](#)

- [巢狀文件的投影](#)

編製陣列索引

自 2020 年 4 月 23 日起，Amazon DocumentDB 現在支援索引大於 2,048 位元組的陣列。數組中的單個項目的限制仍然保持為 2,048 個字節，這與 MongoDB 一致。

如果您要建立新索引，則不需要採取任何動作來利用改進的功能。如果您有現有的索引，您可以刪除索引再重新建立，以利用改進的功能。具有改進功能的最新索引版本為 "v" : 3。

Note

對於生產叢集，刪除索引可能會影響您的應用程式效能。建議您在對生產系統變更時先加以測試並小心進行。此外，重新建立索引所需的時間，將是集合整體資料大小的函數。

您可以使用以下命令查詢索引的版本。

```
db.collection.getIndexes()
```

此操作的輸出將會如下所示。在此輸出中，索引的版本是 "v" : 3，這是最新的索引版本。

```
[
  {
    "v" : 3,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "test.test"
  }
]
```

多金鑰索引

自 2020 年 4 月 23 日起，Amazon DocumentDB 現在支援在同一陣列中建立具有多個鍵的複合索引的功能。

如果您要建立新索引，則不需要採取任何動作來利用改進的功能。如果您有現有的索引，您可以刪除索引再重新建立，以利用改進的功能。具有改進功能的最新索引版本為 "v" : 3。

Note

對於生產叢集，刪除索引可能會影響您的應用程式效能。建議您在對生產系統變更時先加以測試並小心進行。此外，重新建立索引所需的時間，將是集合整體資料大小的函數。

您可以使用以下命令查詢索引的版本。

```
db.collection.getIndexes()
```

此操作的輸出將會如下所示。在此輸出中，索引的版本是 "v" : 3，這是最新的索引版本。

```
[
  {
    "v" : 3,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "test.test"
  }
]
```

字串中的 Null 字元

從 2020 年 6 月 22 日起，Amazon DocumentDB 現在支持字串中的空字元 ('\0')。

以角色為基礎的存取控制

自 2020 年 3 月 26 日起，Amazon DocumentDB 支援內建角色的角色型存取控制 (RBAC)。如需進一步了解，請參閱 [以角色為基礎的存取控制](#)。

\$regex 索引功能

截至 2020 年 6 月 22 日，Amazon DocumentDB 現在支持 \$regex 操作員使用索引的能力。

若要使用與 \$regex 運算子的索引，您必須使用 hint() 命令。使用 hint() 時，您必須指定要套用 \$regex 之欄位的名稱。例如，如果您在欄位 product 有名為 p_1 的索

引，`db.foo.find({product: /^x.*/}).hint({product:1})` 將使用 `p_1` 索引，但 `db.foo.find({product: /^x.*/}).hint("p_1")` 不會使用該索引。您可以使用 `explain()` 命令或使用分析工具的記錄慢速查詢功能來驗證是否已選擇索引。例如 `db.foo.find({product: /^x.*/}).hint("p_1").explain()`。

Note

`hint()` 方法一次只能與一個索引一起使用。

`$regex` 查詢的索引使用已針對使用字首且未指定 `I`、`m` 或 `o` 規則運算式選項的查詢進行最佳化。

將 `$regex` 搭配索引使用時，建議您在有高度選擇性的欄位上建立索引，其中重複值的數目會小於集中文件總數的 1%。例如，如果您的集合包含 100,000 份文件，則只在相同值出現 1000 次或更少的欄位上建立索引。

巢狀文件的投影

有一個功能差異與 Amazon DocumentDB 和 MongoDB 之間的 `$project` 運營商 3.6 版已經在 Amazon DocumentDB 4.0 已經解決，但仍將在 Amazon DocumentDB 3.6 不受支持。

Amazon DocumentDB 3.6 應用投影時只考慮嵌套文檔中的第一個字段，而 MongoDB 3.6 將解析子文檔並將投影應用於每個子文檔以及。

例如：如果投影是 `"a.b.c": 1`，則該行為按預期在 Amazon DocumentDB 和 MongoDB 中工作。但是，如果投影是 `{a:{b:{c:1}}}` 那麼 Amazon DocumentDB 3.6 將只應用於投影，`a` 而不 `b` 是或。 `c` 在 Amazon DocumentDB 4.0，投影 `{a:{b:{c:1}}}` 將被應用到 `ab`，和。 `c`

MongoDB 的功能差異

主題

- [\\$vectorSearch 運算子](#)
- [OpCountersCommand](#)
- [管理員資料庫和集合](#)
- [cursormaxTimeMS](#)
- [explain\(\)](#)
- [欄位名稱限制](#)

- [索引建置](#)
- [在路徑中使用空鍵查找](#)
- [MongoDB API、營運和資料類型](#)
- [mongodump和mongorestore公用事業](#)
- [結果排序](#)
- [可重試的寫入](#)
- [稀疏索引](#)
- [使用 \\$all 表達式內的 \\$elemMatch](#)
- [\\$ne、\\$nin、\\$nor、\\$not\\$exists、和\\$elemMatch索引](#)
- [\\$lookup](#)

\$vectorSearch運算子

Amazon DocumentDB 不支持\$vectorSearch作為獨立運營商。相反，我們支持\$search運營商vectorSearch內部。如需詳細資訊，請參閱 [向量搜索 Amazon DocumentDB](#)。

OpCountersCommand

Amazon 文檔數據庫的OpCountersCommand行為偏離 MongoDB 的，如下所示：`opcounters.command`

- MongoDB 的`opcounters.command`計數除了插入，更新和刪除所有命令，而 Amazon DocumentDB 的OpCountersCommand也排除了命令。`find`
- Amazon DocumentDB 計算內部命令（例如`getCloudWatchMetricsV2`）。OpCountersCommand

管理員資料庫和集合

Amazon DocumentDB 不支持管理員或本地數據庫，也不分別 MongoDB `system.*` 或`startup_log`集合。

cursorMaxTimeMS

在 Amazon DocumentDB 中，為每個`getMore`請求`cursor.maxTimeMS`重置計數器。因此，如果指定了 3000MS，`maxTimeMS`則查詢需要 2800MS，並且每個後續`getMore`請求都需要 300MS，

則光標不會超時。只有當單一作業 (查詢或個別要getMore求) 花費超過指定的作業時，游標才會逾時maxTimeMS。此外，檢查游標執行時間的掃描器會以五 (5) 分鐘的粒度執行。

explain()

Amazon DocumentDB 會在使用分散式容錯、自我修復儲存系統的專用資料庫引擎上模擬 MongoDB 4.0 API。因此，Amazon DocumentDB 和 MongoDB 之間的查詢計劃和輸出explain()可能會有所不同。想要控制其查詢計劃的客戶可以使用 \$hint 運算子強制選取偏好的索引。

欄位名稱限制

Amazon DocumentDB 不支持點「.」在文件欄位名稱中，例如，db.foo.insert({'x.1':1})。

Amazon DocumentDB 也不支持字段名稱中的 \$ 前綴。

例如，嘗試在 Amazon DocumentDB 或 MongoDB 下面的命令：

```
rs0:PRIMARY> db.foo.insert({"a":{"$a":1}})
```

MongoDB 將返回以下內容：

```
WriteResult({ "nInserted" : 1 })
```

Amazon DocumentDB 將返回一個錯誤：

```
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 2,
    "errmsg" : "Document can't have $ prefix field names: $a"
  }
})
```

Note

這種功能差異有一個例外。以 \$ 前綴開頭的以下字段名稱已列入白名單，並且可以在 Amazon DocumentDB 中成功使用：\$ ID，\$ 參考和 \$ 分貝。

索引建置

Amazon DocumentDB 只允許在任何給定時間在集合上建立一個索引。無論是在前景還是背景中。當索引建置目前正在進行時，如果在相同的集合上進行如 `createIndex()` 或 `dropIndex()` 等操作，最新嘗試進行的操作將會失敗。

默認情況下，Amazon DocumentDB 和 MongoDB 4.0 版本中的索引構建發生在後台。MongoDB 版本 4.2，如果指定為 `createIndexes` 或其外殼助手和，則忽略後台索引構建選項。`createIndex()` `createIndexes()`

存留時間 (TTL) 索引會在索引建立完成後開始將到期的文件。

在路徑中使用空鍵查找

當您查找包含空字符串作為路徑一部分的鍵 (例如，`x..b`) `x.`，並且該對象在數組中具有空字符串鍵路徑 (例如 `{ "x" : [{ "" : 10 }, { "b" : 20 }] }`) 時，Amazon DocumentDB 將返回不同的結果，而不是在 MongoDB 中運行相同的查找。

在 MongoDB 中，當空字符串鍵不在路徑末尾查找時，空鍵路徑在數組中查找按預期工作。但是，當空字符串鍵位於路徑末尾查找時，它不會查看數組。

但是在 Amazon DocumentDB 中，只會讀取陣列中的第一個元素，因為 `getArrayIndexFromKeyString` 將空字串轉換為 0，所以字串鍵查找會被視為陣列索引查找。

MongoDB API、營運和資料類型

Amazon DocumentDB 是與 MongoDB 3.6 和 4.0 的 API 兼容。如需支援功能的 up-to-date 清單，請參閱 [支援的 MongoDB API、營運和資料類型](#)。

mongodump和mongoexport公用事業

Amazon DocumentDB 不支援管理員資料庫，因此在使用或公用程式時不會傾印或還原管理員資料庫。mongodump mongoexport當您使用 Amazon DocumentDB 建立新資料庫時mongoexport，除了還原操作之外，還需要重新建立使用者角色。

Note

對於 Amazon DocumentDB，我們推薦 MongoDB 數據庫工具，並包括版本 100.6.1。您可以訪問 MongoDB 的數據庫工具下載 [這裡](#)。

結果排序

Amazon DocumentDB 不保證結果集的隱式結果排序順序。為確保結果集的排序，請使用 `sort()` 明確指定排序順序。

以下範例會根據庫存欄位以遞減順序排序庫存集合中的項目。

```
db.inventory.find().sort({ stock: -1 })
```

使用 `$sort` 彙總階段時，除非階段是彙總管線中的最後一個 `$sort` 階段，否則不會保留排序順序。將 `$sort` 彙總階段與彙總階段結合使用時，`$sort` 彙總階段僅適用於 `$first` 和 `$last` 累加器。`$group` 在 Amazon DocumentDB 4.0 中，增加了支持 `$push` 以尊重上一 `$sort` 階段的排序順序。

可重試的寫入

從與 MongoDB 4.2 相容的驅動程式開始使用，可重試的寫入為預設啟用。不過，Amazon DocumentDB 目前不支援可重試寫入。功能差異會在類似下列錯誤訊息中以資訊清單列出。

```
{"ok":0,"errmsg":"Unrecognized field: 'txnNumber',"code":9,"name":"MongoError"}
```

可重試寫入可透過連接字串停用 (例如，`MongoClient("mongodb://my.mongodb.cluster/db?retryWrites=false")`) 或 `MongoClient` 建構函式的關鍵字引數 (例如，`MongoClient("mongodb://my.mongodb.cluster/db", retryWrites=False)`)

以下是透過連線字串停用可重試寫入的 Python 範例。

```
client =
    pymongo.MongoClient('mongodb://
<username>:<password>@docdb-2019-03-17-16-49-12.cluster-ccuszbx3pn5e.us-
east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0',w='majority',j=True,retryWrites=False)
```

稀疏索引

若要使用您在查詢中建立的稀疏索引，您必須在涵蓋索引的欄位上使用 `$exists` 子句。如果省略 `$exists`，Amazon DocumentDB 不會使用稀疏索引。

以下是範例。

```
db.inventory.count({ "stock": { $exists: true } })
```

對於稀疏、多索引鍵索引，Amazon DocumentDB 不支援唯一索引鍵條件約束，如果查詢文件會導致一組值，且只遺失索引欄位的子集。例如，若輸入為 "a" : [{ "b" : 2 }, { "c" : 1 }] 則不支援 `createIndex({"a.b" : 1 }, { unique : true, sparse :true })`，因為 "a.c" 會儲存在索引中。

使用 \$all 表達式內的 \$elemMatch

Amazon DocumentDB 目前不支持在表達式中使用 `$elemMatch` 運算符。`$all` 您可以如下所示搭配 `$elemMatch` 使用 `$and`，解決問題。

原始操作：

```
db.col.find({
  qty: {
    $all: [
      { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } } },
      { "$elemMatch": { num: 40, size: "XL" } }
    ]
  }
})
```

更新的操作：

```
db.col.find({
  $and: [
    { qty: { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } } } },
    { qty: { "$elemMatch": { qty: 40, size: "XL" } } }
  ]
})
```

\$ne、\$nin、\$nor、\$not\$exists、和\$elemMatch索引

Amazon DocumentDB 目前不支援使用索引與 `$ne`、`$nin`、`$nor` 和 `$distinct` 運算子搭配使用的功能。`$not $exists` 因此，使用這些運算子將導致收集掃描。在使用其中一個運算子之前執行篩選或比對會減少需要掃描的資料量，進而改善效能。

Amazon DocumentDB 增加了對亞 Amazon DocumentDB 5.0 和彈性集群中的\$elemMatch操作員進行索引掃描的支持。如果僅查詢篩選具有一個層級的篩選器，則支援索引掃描，但如果包含巢狀\$elemMatch查詢，則不支援索引掃描。\$elemMatch

\$elemMatch支援索引掃描 Amazon DocumentDB 狀：

```
db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
```

\$elemMatch不支援索引掃描的 Amazon DocumentDB 狀：

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e": "abc"} }} } })
```

\$lookup

Amazon DocumentDB 支援執行相等比對的功能 (例如，左外部聯結)，但不支援不相關的子查詢。

利用索引 \$lookup

您現在可以使用\$lookup階段運算子的索引。根據您的使用案例，您可以使用多種索引演算法來最佳化效能。本節將說明不同的索引演算法，\$lookup並協助您選擇最適合您工作負載的索引演算法。

根據預設，Amazon DocumentDB 在使用時會利用雜湊演算法，並allowDiskUse:false在使用時對合併進行排序。allowDiskUse:true對於某些使用案例，可能需要強制查詢最佳化工具使用不同的演算法。以下是\$lookup聚合運算符可以利用的不同索引算法：

- 巢狀迴圈：如果外部集合小於 1 GB，而外部集合中的欄位具有索引，巢狀迴圈計畫通常對工作負載有益。如果使用巢狀迴圈演算法，說明計畫會將階段顯示為NESTED_LOOP_LOOKUP。
- 排序合併：如果外部集合在查詢中使用的欄位上沒有索引，且工作資料集不適合記憶體，則排序合併計畫通常對工作負載有益。如果使用排序合併演算法，說明計畫會將階段顯示為SORT_LOOKUP。
- 雜湊：如果外部集合小於 1GB，且工作資料集適合記憶體，雜湊計畫通常對工作負載有益。如果正在使用雜湊演算法，說明計畫會將階段顯示為HASH_LOOKUP。

您可以在查詢上使用 explain 來識別正在用於\$lookup運算子的索引演算法。下面是一個例子。

```
db.localCollection.explain().
aggregate( [
  {
    $lookup:
```

```

        {
            from: "foreignCollection",
            localField: "a",
            foreignField: "b",
            as: "joined"
        }
    ]
}

output
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "test.localCollection",
        "winningPlan" : {
            "stage" : "SUBSCAN",
            "inputStage" : {
                "stage" : "SORT_AGGREGATE",
                "inputStage" : {
                    "stage" : "SORT",
                    "inputStage" : {
                        "stage" : "NESTED_LOOP_LOOKUP",
                        "inputStages" : [
                            {
                                "stage" : "COLLSCAN"
                            },
                            {
                                "stage" : "FETCH",
                                "inputStage" : {
                                    "stage" : "COLLSCAN"
                                }
                            }
                        ]
                    }
                }
            }
        }
    },
    "serverInfo" : {
        "host" : "devbox-test",
        "port" : 27317,
        "version" : "3.6.0"
    },
    "ok" : 1
}

```

```
}

```

作為使用方法的替代方`explain()`法，您可以使用效能分析工具來檢閱您使用運算子時所使用的`$lookup`演算法。如需效能分析工具的詳細資訊，請參閱。[分析亞馬遜文檔數據庫操作](#)

使用 `planHint`

如果您想要強制查詢最佳化工具使用不同的索引演算法`$lookup`，您可以使用`planHint`。若要這麼做，請使用彙總階段選項中的註解來強制執行不同的計劃。下面是評論的語法的一個例子：

```
comment : {
  comment : "<string>",
  lookupStage : { planHint : "SORT" | "HASH" | "NESTED_LOOP" }
}
```

下面是使用強制查詢優化器使用HASH索引算法的一個例子：`planHint`

```
db.foo.aggregate(
  [
    {
      $lookup:
      {
        from: "foo",
        localField: "_id",
        foreignField: "_id",
        as: "joined"
      },
    }
  ],
  {
    comment : "{ \"lookupStage\" : { \"planHint\" : \"HASH\" } }"
```

若要測試哪種演算法最適合您的工作負載，您可以在修改索引演算`explain`法 (即HASH//SORT//NESTED_LOOP) 時，使用方法的`executionStats`參數來測量`$lookup`階段的執行時間。

下列範例顯示如`executionStats`何使用SORT演算法來測量`$lookup`階段的執行時間。

```
db.foo.explain("executionStats").aggregate(
  [
    {
```

```
$lookup:
{
  from: "foo",
  localField: "_id",
  foreignField: "_id",
  as: "joined"
},
}
],
{
  comment : "{ \\\"lookupStage\\\" : { \\\"planHint\\\": \\\"SORT\\\" }}"
```


支援的 MongoDB API、營運和資料類型

Amazon DocumentDB (與 MongoDB 相容性) 是一種快速、可擴展、高度可用且全受管的文件資料庫服務，可支援 MongoDB 工作負載。Amazon DocumentDB 是與 MongoDB 3.6，4.0 和 5.0 的 API 兼容。本區段列出支援的功能。如需使用 MongoDB API 和驅動程式的支援，請參閱 MongoDB 社群論壇。如需使用 Amazon DocumentDB 服務的支援，請聯絡適當的 AWS 支援團隊。有關 Amazon DocumentDB 和 MongoDB 之間的功能差異，請參閱 [功能差異：Amazon DocumentDB 和 MongoDB](#)

不支援 MongoDB 命令和運算子，這些命令僅供內部使用或不適用於全受管服務，且未包含在支援的功能清單中。

自推出以來，我們已新增超過 50 項額外功能，並會繼續向客戶提供他們所需的功能。如需最近推出的相關資訊，請參閱 [Amazon DocumentDB](#) 公告。

如果您希望我們建置某項功能不受支援，請將包含您的 accountID、要求的功能和使用案例的電子郵件傳送給 [Amazon DocumentDB](#) 服務團隊，讓我們知道。

主題

- [資料庫命令](#)
- [查詢和投影運算子](#)
- [更新運算子](#)
- [Geospatial \(地理空間\)](#)
- [游標方法](#)
- [彙總管道運算子](#)
- [資料類型](#)
- [索引和索引屬性](#)

資料庫命令

主題

- [管理命令](#)
- [聚合](#)
- [身分驗證](#)
- [診斷命令](#)

- [查詢和寫入操作](#)
- [角色管理命令](#)
- [階段作業命](#)
- [使用者管理](#)
- [分片命令](#)

管理命令

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|------------------------------|-----|-----|-----|------|
| Capped Collections | 否 | 否 | 否 | 否 |
| cloneCollectionAs封頂 | 否 | 否 | 否 | 否 |
| collMod | 部分 | 部分 | 部分 | 是 |
| collMod : expireAfterSeconds | 是 | 是 | 是 | 否 |
| convertToCapped | 否 | 否 | 否 | 否 |
| copydb | 否 | 否 | 否 | 否 |
| 建立 | 是 | 是 | 是 | 是 |
| createView | 否 | 否 | 否 | 否 |
| createIndexes | 是 | 是 | 是 | 是 |
| currentOp | 是 | 是 | 是 | 是 |
| drop | 是 | 是 | 是 | 是 |
| dropDatabase | 是 | 是 | 是 | 是 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|------------------|-----|-----|-----|------|
| dropIndexes | 是 | 是 | 是 | 是 |
| filemd5 | 否 | 否 | 否 | 否 |
| killCursors | 是 | 是 | 是 | 是 |
| killOp | 是 | 是 | 是 | 是 |
| 名單收集 * | 是 | 是 | 是 | 是 |
| listDatabases | 是 | 是 | 是 | 是 |
| listIndexes | 是 | 是 | 是 | 是 |
| reIndex | 否 | 否 | 否 | 否 |
| renameCollection | 是 | 是 | 是 | 否 |

* 不支援篩選選項中的type金鑰。

聚合

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-----------|-----|-----|-----|------|
| aggregate | 是 | 是 | 是 | 是 |
| count | 是 | 是 | 是 | 是 |
| distinct | 是 | 是 | 是 | 是 |
| mapReduce | 否 | 否 | 否 | 否 |

身分驗證

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------|-----|-----|-----|------|
| authenticate | 是 | 是 | 是 | 是 |
| 登出 | 是 | 是 | 是 | 是 |

診斷命令

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------------------|----------|----------|----------|------|
| buildInfo | 是 | 是 | 是 | 是 |
| collStats | 是 | 是 | 是 | 是 |
| connPoolStats | 否 | 否 | 否 | 否 |
| connectionStatus | 是 | 是 | 是 | 是 |
| dataSize | 是 | 是 | 是 | 是 |
| dbHash | 否 | 否 | 否 | 否 |
| dbStats | 是 | 是 | 是 | 是 |
| explain | 是 | 是 | 是 | 是 |
| explain: executionStats | 是 | 是 | 是 | 是 |
| 功能 | 否 | 否 | 否 | 否 |
| hostInfo | 是 | 是 | 是 | 是 |
| listCommands | 是 | 是 | 是 | 是 |
| profiler | <u>是</u> | <u>是</u> | <u>是</u> | 否 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------|-----|-----|-----|------|
| serverStatus | 是 | 是 | 是 | 是 |
| top | 是 | 是 | 是 | 是 |

查詢和寫入操作

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-------------------------|----------|----------|----------|------|
| 刪除 | 是 | 是 | 是 | 是 |
| find | 是 | 是 | 是 | 是 |
| findAndModify | 是 | 是 | 是 | 是 |
| getLastError | 否 | 否 | 否 | 否 |
| getMore | 是 | 是 | 是 | 是 |
| getPrevError | 否 | 否 | 否 | 否 |
| insert | 是 | 是 | 是 | 是 |
| parallelCollection Scan | 否 | 否 | 否 | 否 |
| resetError | 否 | 否 | 否 | 否 |
| update | 是 | 是 | 是 | 是 |
| Change streams | <u>是</u> | <u>是</u> | <u>是</u> | 否 |
| GridFS | 否 | 否 | 否 | 否 |
| ReplaceOne | 是 | 是 | 是 | 是 |

角色管理命令



C 3 4 5 引
作
畫
集

c 是 是 是 否
e

c 是 是 是 否

c 是 是 是 否
k a
t

c 是 是 是 否
s
f
é

r 是 是 是 否
v
c
f
é

r 是 是 是 否
e
f
é

u 是 是 是 否
e

r 是 是 是 否

階段作業命

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------------------|-----|-----|-----|------|
| 中止交易 | 否 | 是 | 是 | 否 |
| commitTransaction | 否 | 是 | 是 | 否 |
| 工作階段 | 否 | 否 | 否 | 否 |
| killAllSessions | 否 | 是 | 是 | 否 |
| killAllSessionsByPattern | 否 | 否 | 否 | 否 |
| 杀戮会话 | 否 | 是 | 是 | 否 |
| 重新整理作業 | 否 | 否 | 否 | 否 |
| 開始工作階 | 否 | 是 | 是 | 否 |

使用者管理

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------------------|-----|-----|-----|------|
| createUser | 是 | 是 | 是 | 是 |
| dropAllUsersFromDatabase | 是 | 是 | 是 | 是 |
| dropUser | 是 | 是 | 是 | 是 |
| grantRolesTo使用者 | 是 | 是 | 是 | 是 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------------|-----|-----|-----|------|
| revokeRolesFrom使用者 | 是 | 是 | 是 | 是 |
| updateUser | 是 | 是 | 是 | 是 |
| userInfo | 是 | 是 | 是 | 是 |

分片命令

| Command | 彈性叢集 |
|--------------------------|------|
| abortReshardCollection | 否 |
| 添加碎片 | 否 |
| addShardTo區域 | 否 |
| balancerCollectionStatus | 否 |
| 平衡器啟動 | 否 |
| 平衡器狀態 | 否 |
| 平衡器 | 否 |
| checkShardingIndex | 否 |
| clearJumboFlag | 否 |
| cleanupOrphaned | 否 |
| cleanupReshardCollection | 否 |
| commitReshardCollection | 否 |
| 啟用分享 | 是 |
| flushRouterConfig | 否 |

| Command | 彈性叢集 |
|-------------------------|------|
| getShardMap | 否 |
| getShardVersion | 否 |
| ISDBGRID | 否 |
| 列表碎片 | 否 |
| 媒體鑰匙 | 否 |
| 移動塊 | 否 |
| 运动初級 | 否 |
| 合併塊 | 否 |
| refineCollectionShard關鍵 | 否 |
| 移除硬體 | 否 |
| removeShardFrom區域 | 否 |
| 重新碎片收集 | 否 |
| setAllowMigrations | 否 |
| setShardVersion | 否 |
| 鯊魚收藏 | 是 |
| 分享狀態 | 否 |
| 分割 | 否 |
| 分割向量 | 否 |
| 未分片 | 否 |
| updateZoneKey範圍 | 否 |

查詢和投影運算子

主題

- [陣列運算子](#)
- [位元運算子](#)
- [註解運算子](#)
- [比較運算子](#)
- [元素運算子](#)
- [評估查詢運算子](#)
- [邏輯運算子](#)
- [投影運算子](#)

陣列運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-------------|-----|-----|-----|------|
| \$all | 是 | 是 | 是 | 是 |
| \$elemMatch | 是 | 是 | 是 | 是 |
| \$size | 是 | 是 | 是 | 是 |

位元運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------|-----|-----|-----|------|
| \$bitsAllSet | 是 | 是 | 是 | 是 |
| \$bitsAnySet | 是 | 是 | 是 | 是 |
| \$bitsAllClear | 是 | 是 | 是 | 是 |
| \$bitsAnyClear | 是 | 是 | 是 | 是 |

註解運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-----------|-----|-----|-----|------|
| \$comment | 是 | 是 | 是 | 是 |

比較運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$eq | 是 | 是 | 是 | 是 |
| \$gt | 是 | 是 | 是 | 是 |
| \$gte | 是 | 是 | 是 | 是 |
| \$lt | 是 | 是 | 是 | 是 |
| \$lte | 是 | 是 | 是 | 是 |
| \$ne | 是 | 是 | 是 | 是 |
| \$in | 是 | 是 | 是 | 是 |
| \$nin | 是 | 是 | 是 | 是 |

元素運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------|-----|-----|-----|------|
| \$exists | 是 | 是 | 是 | 是 |
| \$type | 是 | 是 | 是 | 是 |

評估查詢運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|------------------------------|-----|-----|-----|------|
| \$expr | 否 | 否 | 否 | 否 |
| \$jsonSchema | 否 | 是 | 是 | 否 |
| \$mod | 是 | 是 | 是 | 是 |
| \$regex | 是 | 是 | 是 | 是 |
| \$text | 否 | 否 | 是 | 否 |
| \$where | 否 | 否 | 否 | 否 |

邏輯運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$or | 是 | 是 | 是 | 是 |
| \$and | 是 | 是 | 是 | 是 |
| \$not | 是 | 是 | 是 | 是 |
| \$nor | 是 | 是 | 是 | 是 |

投影運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-------------|-----|-----|-----|------|
| \$ | 是 | 是 | 是 | 是 |
| \$elemMatch | 是 | 是 | 是 | 是 |
| \$meta | 否 | 否 | 是 | 否 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$slice | 是 | 是 | 是 | 是 |

更新運算子

主題

- [陣列運算子](#)
- [位元運算子](#)
- [欄位運算子](#)
- [更新修改程式](#)

陣列運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|------------------|-----|-----|-----|------|
| \$ | 是 | 是 | 是 | 是 |
| \$[] | 是 | 是 | 是 | 是 |
| \$[<identifier>] | 是 | 是 | 是 | 是 |
| \$addToSet | 是 | 是 | 是 | 是 |
| \$pop | 是 | 是 | 是 | 是 |
| \$pullAll | 是 | 是 | 是 | 是 |
| \$pull | 是 | 是 | 是 | 是 |
| \$push | 是 | 是 | 是 | 是 |

位元運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$bit | 是 | 是 | 是 | 是 |

欄位運算子

| 運算子 | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------------|-----|-----|-----|------|
| \$inc | 是 | 是 | 是 | 是 |
| \$mul | 是 | 是 | 是 | 是 |
| \$rename | 是 | 是 | 是 | 是 |
| \$setOnInsert | 是 | 是 | 是 | 是 |
| \$set | 是 | 是 | 是 | 是 |
| \$unset | 是 | 是 | 是 | 是 |
| \$min | 是 | 是 | 是 | 是 |
| \$max | 是 | 是 | 是 | 是 |
| \$currentDate | 是 | 是 | 是 | 是 |

更新修改程式

| 運算子 | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$each | 是 | 是 | 是 | 是 |
| \$slice | 是 | 是 | 是 | 是 |
| \$sort | 是 | 是 | 是 | 是 |

| 運算子 | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|------------|-----|-----|-----|------|
| \$position | 是 | 是 | 是 | 是 |

Geospatial (地理空間)

幾何指定元

| 查詢選取器 | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------|-----|-----|-----|------|
| \$box | 否 | 否 | 否 | 否 |
| \$center | 否 | 否 | 否 | 否 |
| \$centerSphere | 否 | 否 | 否 | 否 |
| \$nearSphere | 是 | 是 | 是 | 否 |
| \$geometry | 是 | 是 | 是 | 否 |
| \$maxDistance | 是 | 是 | 是 | 否 |
| \$minDistance | 是 | 是 | 是 | 否 |
| \$polygon | 否 | 否 | 否 | 否 |
| \$uniqueDocs | 否 | 否 | 否 | 否 |

查詢選取器

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-----------------|-----|-----|-----|------|
| \$geoIntersects | 是 | 是 | 是 | 否 |
| \$geoWithin | 是 | 是 | 是 | 否 |
| \$near | 否 | 否 | 否 | 否 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------|-----|-----|-----|------|
| \$nearSphere | 是 | 是 | 是 | 否 |
| \$polygon | 否 | 否 | 否 | 否 |
| \$uniqueDocs | 否 | 否 | 否 | 否 |

游標方法

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------------|-----|-----|-----|------|
| cursor.batchSize() | 是 | 是 | 是 | 是 |
| cursor.close() | 是 | 是 | 是 | 是 |
| cursor.isClosed() | 是 | 是 | 是 | 是 |
| cursor.collation() | 否 | 否 | 否 | 否 |
| cursor.comment() | 是 | 是 | 是 | 是 |
| cursor.count() | 是 | 是 | 是 | 是 |
| cursor.explain() | 是 | 是 | 是 | 否 |
| cursor.forEach() | 是 | 是 | 是 | 是 |
| cursor.hasNext() | 是 | 是 | 是 | 是 |
| cursor.hint() | 是 | 是 | 是 | 是 * |
| cursor.isExhausted() | 是 | 是 | 是 | 否 |
| cursor.itcount() | 是 | 是 | 是 | 否 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------------|-----|-----|-----|------|
| cursor.limit() | 是 | 是 | 是 | 否 |
| cursor.map() | 是 | 是 | 是 | 否 |
| cursor.maxScan() | 是 | 是 | 是 | 否 |
| cursor.maxTimeMS() | 是 | 是 | 是 | 否 |
| cursor.max() | 否 | 否 | 否 | 否 |
| cursor.min() | 否 | 否 | 否 | 否 |
| cursor.next() | 是 | 是 | 是 | 是 |
| 游標。noCursorTimeout() | 否 | 否 | 否 | 否 |
| 游標。objsLeftInBatch() | 是 | 是 | 是 | 否 |
| cursor.pretty() | 是 | 是 | 是 | 否 |
| cursor.readConcern() | 是 | 是 | 是 | 否 |
| cursor.readPref() | 是 | 是 | 是 | 否 |
| cursor.returnKey() | 否 | 否 | 否 | 否 |
| 游標。showRecordId() | 否 | 否 | 否 | 否 |
| cursor.size() | 是 | 是 | 是 | 否 |
| cursor.skip() | 是 | 是 | 是 | 否 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------------------------|-----|-----|-----|------|
| <code>cursor.sort()</code> | 是 | 是 | 是 | 否 |
| <code>cursor.tailable()</code> | 否 | 否 | 否 | 否 |
| <code>cursor.toArray()</code> | 是 | 是 | 是 | 否 |

* 索引hint運算式支援索引。例如 `db.foo.find().hint({x:1})`。

彙總管道運算子

主題

- [累加器表達式](#)
- [算術運算子](#)
- [陣列運算子](#)
- [布林值運算子](#)
- [比較運算子](#)
- [條件式表達式運算子](#)
- [日期類型運算子](#)
- [資料大小運算子](#)
- [日期運算子](#)
- [常值運算子](#)
- [合併運算子](#)
- [自然操作](#)
- [設定運算子](#)
- [階段運算子](#)
- [字串運算子](#)
- [系統變數](#)
- [文字搜尋運算子](#)
- [類型轉換運算子](#)
- [變數運算子](#)

- [雜項運算子](#)

累加器表達式

| 表達式 | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------|-----|-----|-----|------|
| \$sum | 是 | 是 | 是 | 是 |
| \$avg | 是 | 是 | 是 | 是 |
| \$first | 是 | 是 | 是 | 是 |
| \$last | 是 | 是 | 是 | 是 |
| \$max | 是 | 是 | 是 | 是 |
| \$min | 是 | 是 | 是 | 是 |
| \$push | 是 | 是 | 是 | 是 |
| \$addToSet | 是 | 是 | 是 | 是 |
| \$stdDevPop | 否 | 否 | 否 | 否 |
| \$stdDevSamp | 否 | 否 | 否 | 否 |
| \$累加器 | - | - | 否 | 否 |
| \$count | - | - | 否 | 否 |

算術運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$abs | 是 | 是 | 是 | 是 |
| \$add | 是 | 是 | 是 | 是 |
| \$ceil | 否 | 是 | 是 | 是 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|------------|-----|-----|-----|------|
| \$divide | 是 | 是 | 是 | 是 |
| \$exp | 否 | 是 | 是 | 是 |
| \$floor | 否 | 是 | 是 | 是 |
| \$ln | 否 | 是 | 是 | 是 |
| \$log | 否 | 是 | 是 | 是 |
| \$log10 | 否 | 是 | 是 | 是 |
| \$mod | 是 | 是 | 是 | 是 |
| \$multiply | 是 | 是 | 是 | 是 |
| \$pow | 否 | 否 | 否 | 否 |
| \$sqrt | 否 | 是 | 是 | 是 |
| \$subtract | 是 | 是 | 是 | 是 |
| \$trunc | 否 | 否 | 否 | 否 |
| \$ 一輪 | - | - | 否 | 否 |

陣列運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-----------------|-----|-----|-----|------|
| \$arrayElemAt | 是 | 是 | 是 | 是 |
| \$arrayToObject | 是 | 是 | 是 | 是 |
| \$concatArrays | 是 | 是 | 是 | 是 |
| \$filter | 是 | 是 | 是 | 是 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-----------------|-----|-----|-----|------|
| \$indexOfArray | 是 | 是 | 是 | 是 |
| \$isArray | 是 | 是 | 是 | 是 |
| \$objectToArray | 是 | 是 | 是 | 是 |
| \$range | 是 | 是 | 是 | 是 |
| \$reverseArray | 是 | 是 | 是 | 是 |
| \$reduce | 是 | 是 | 是 | 是 |
| \$size | 是 | 是 | 是 | 是 |
| \$slice | 是 | 是 | 是 | 是 |
| \$zip | 是 | 是 | 是 | 是 |
| \$in | 是 | 是 | 是 | 是 |
| \$first | - | - | 否 | 否 |
| \$last | - | - | 否 | 否 |

布林值運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$and | 是 | 是 | 是 | 是 |
| \$or | 是 | 是 | 是 | 是 |
| \$not | 是 | 是 | 是 | 是 |

比較運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$cmp | 是 | 是 | 是 | 是 |
| \$eq | 是 | 是 | 是 | 是 |
| \$gt | 是 | 是 | 是 | 是 |
| \$gte | 是 | 是 | 是 | 是 |
| \$lt | 是 | 是 | 是 | 是 |
| \$lte | 是 | 是 | 是 | 是 |
| \$ne | 是 | 是 | 是 | 是 |

條件式表達式運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------|-----|-----|-----|------|
| \$cond | 是 | 是 | 是 | 是 |
| \$ifNull | 是 | 是 | 是 | 是 |
| \$switch | 否 | 否 | 否 | 否 |

日期類型運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$type | 是 | 是 | 是 | 是 |

資料大小運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------|-----|-----|-----|------|
| \$ 二進制大小 | - | - | 否 | 否 |
| B 尺寸 | - | - | 否 | 否 |

日期運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------|-----|-----|-----|------|
| \$ 日期添加 | 否 | 否 | 是 | 是 |
| \$ 日期減法 | 否 | 否 | 是 | 是 |
| \$dayOfYear | 是 | 是 | 是 | 是 |
| \$dayOfMonth | 是 | 是 | 是 | 是 |
| \$dayOfWeek | 是 | 是 | 是 | 是 |
| \$year | 是 | 是 | 是 | 是 |
| \$month | 是 | 是 | 是 | 是 |
| \$week | 是 | 是 | 是 | 是 |
| \$hour | 是 | 是 | 是 | 是 |
| \$minute | 是 | 是 | 是 | 是 |
| \$second | 是 | 是 | 是 | 是 |
| \$millisecond | 是 | 是 | 是 | 是 |
| \$dateToString | 是 | 是 | 是 | 是 |
| 美元isoDayOf週 | 是 | 是 | 是 | 是 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|------------------|-----|-----|-----|------|
| \$isoWeek | 是 | 是 | 是 | 是 |
| \$dateFromParts | 否 | 否 | 否 | 否 |
| \$dateToParts | 否 | 否 | 否 | 否 |
| \$dateFromString | 是 | 是 | 是 | 是 |
| \$isoWeekYear | 是 | 是 | 是 | 是 |
| \$ 數據運行 | - | - | 否 | 否 |
| \$ 數據加 | - | - | 否 | 否 |

常值運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-----------|-----|-----|-----|------|
| \$literal | 是 | 是 | 是 | 是 |

合併運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------|-----|-----|-----|------|
| \$mergeObjects | 是 | 是 | 是 | 是 |

自然操作

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$ 自然 | 是 | 是 | 是 | 是 |

設定運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-------------------|-----|-----|-----|------|
| \$setEquals | 是 | 是 | 是 | 是 |
| \$setIntersection | 是 | 是 | 是 | 是 |
| \$setUnion | 是 | 是 | 是 | 是 |
| \$setDifference | 否 | 是 | 是 | 是 |
| \$setIsSubset | 是 | 是 | 是 | 是 |
| \$anyElementTrue | 否 | 是 | 是 | 是 |
| \$allElementsTrue | 否 | 是 | 是 | 是 |

階段運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-------------|-----|-----|-----|------|
| \$collStats | 否 | 否 | 否 | 否 |
| \$project | 是 | 是 | 是 | 是 |
| \$match | 是 | 是 | 是 | 是 |
| \$redact | 是 | 是 | 是 | 是 |
| \$limit | 是 | 是 | 是 | 是 |
| \$skip | 是 | 是 | 是 | 是 |
| \$unwind | 是 | 是 | 是 | 是 |
| \$group | 是 | 是 | 是 | 是 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------------------|-----|-----|-----|------|
| \$sample | 是 | 是 | 是 | 是 |
| \$sort | 是 | 是 | 是 | 是 |
| \$geoNear | 是 | 是 | 是 | 否 |
| \$lookup | 是 | 是 | 是 | 是 |
| \$out | 是 | 是 | 是 | 否 |
| \$indexStats | 是 | 是 | 是 | 是 |
| \$facet | 否 | 否 | 否 | 否 |
| \$bucket | 否 | 否 | 否 | 否 |
| \$bucketAuto | 否 | 否 | 否 | 否 |
| \$sortByCount | 否 | 否 | 否 | 否 |
| \$addFields | 是 | 是 | 是 | 是 |
| \$replaceRoot | 是 | 是 | 是 | 是 |
| \$count | 是 | 是 | 是 | 是 |
| \$currentOp | 是 | 是 | 是 | 是 |
| \$listLocalSessions | 否 | 否 | 否 | 否 |
| \$listSessions | 否 | 否 | 否 | 否 |
| \$graphLookup | 否 | 否 | 否 | 否 |
| \$ 合併 | - | - | 否 | 否 |
| \$planCacheStats | - | - | 否 | 否 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-------------------|-----|-----|-----|------|
| \$setWindowFields | - | - | 否 | 否 |
| \$ 聯盟 | - | - | 否 | 否 |
| \$unset | - | - | 否 | 否 |

字串運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------|-----|-----|-----|------|
| \$concat | 是 | 是 | 是 | 是 |
| \$indexOfBytes | 是 | 是 | 是 | 是 |
| \$indexOfCP | 是 | 是 | 是 | 是 |
| \$ 修剪 | 否 | 否 | 否 | 否 |
| \$ R 修剪 | 否 | 否 | 否 | 否 |
| \$split | 是 | 是 | 是 | 是 |
| \$strcasecmp | 是 | 是 | 是 | 是 |
| \$strLenBytes | 是 | 是 | 是 | 是 |
| \$strLenCP | 是 | 是 | 是 | 是 |
| \$substr | 是 | 是 | 是 | 是 |
| \$substrBytes | 是 | 是 | 是 | 是 |
| \$substrCP | 是 | 是 | 是 | 是 |
| \$toLower | 是 | 是 | 是 | 是 |
| \$toUpper | 是 | 是 | 是 | 是 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------|-----|-----|-----|------|
| \$ 修剪 | 否 | 否 | 否 | 否 |
| \$ 雷吉斯查找 | - | - | 否 | 否 |
| \$regexFindAll | - | - | 否 | 否 |
| \$ 正則表達式匹 配 | - | - | 否 | 否 |
| \$ 替換一 | - | - | 否 | 否 |
| \$ 替換所有 | - | - | 否 | 否 |

系統變數

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-------------|-----|-----|-----|------|
| \$\$CURRENT | 否 | 否 | 否 | 否 |
| \$\$DESCEND | 是 | 是 | 是 | 是 |
| \$\$KEEP | 是 | 是 | 是 | 是 |
| \$\$PRUNE | 是 | 是 | 是 | 是 |
| \$\$REMOVE | 否 | 否 | 否 | 否 |
| \$\$ROOT | 是 | 是 | 是 | 是 |

文字搜尋運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$ 搜索 | 否 | 否 | 是 | 否 |
| \$meta | 否 | 否 | 是 | 否 |

類型轉換運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|--------------|-----|-----|-----|------|
| \$ 轉換 | 否 | 是 | 是 | 是 |
| \$ 平板醇 | 否 | 是 | 是 | 是 |
| \$ 迄今 | 否 | 是 | 是 | 是 |
| \$ 到十進制 | 否 | 是 | 是 | 是 |
| \$ 到翻倍 | 否 | 是 | 是 | 是 |
| \$ 一點 | 否 | 是 | 是 | 是 |
| \$ 托隆 | 否 | 是 | 是 | 是 |
| \$toObjectId | 否 | 是 | 是 | 是 |
| \$ toString | 否 | 是 | 是 | 是 |
| \$ 伊斯號 | - | - | 否 | 否 |

變數運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$map | 是 | 是 | 是 | 是 |
| \$let | 是 | 是 | 是 | 是 |

雜項運算子

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| \$ 兰特 | - | - | 否 | 否 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------|-----|-----|-----|------|
| \$ 樣本 | - | - | 否 | 否 |
| \$ 獲取菲爾德 | - | - | 否 | 否 |

資料類型

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|-------------|-----|-----|-----|------|
| Double | 是 | 是 | 是 | 是 |
| 字串 | 是 | 是 | 是 | 是 |
| 物件 | 是 | 是 | 是 | 是 |
| 陣列 | 是 | 是 | 是 | 是 |
| 二進位資料 | 是 | 是 | 是 | 是 |
| ObjectId | 是 | 是 | 是 | 是 |
| Boolean | 是 | 是 | 是 | 是 |
| 日期 | 是 | 是 | 是 | 是 |
| Null | 是 | 是 | 是 | 是 |
| 32 位整數 | 是 | 是 | 是 | 是 |
| 時間戳記 | 是 | 是 | 是 | 是 |
| 64 位元整數 (長) | 是 | 是 | 是 | 是 |
| MinKey | 是 | 是 | 是 | 是 |
| MaxKey | 是 | 是 | 是 | 是 |
| Decimal128 | 是 | 是 | 是 | 是 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------------------------|-----|-----|-----|------|
| Regular expression (常規表達式) | 是 | 是 | 是 | 是 |
| JavaScript | 否 | 否 | 否 | 否 |
| JavaScript(有範圍) | 否 | 否 | 否 | 否 |
| 未定義 | 否 | 否 | 否 | 否 |
| 符號 | 否 | 否 | 否 | 否 |
| DBPointer | 否 | 否 | 否 | 否 |

索引和索引屬性

主題

- [索引](#)
- [索引屬性](#)

索引

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|----------|-----|-----|-----|------|
| 單一欄位索引 | 是 | 是 | 是 | 是 |
| 複合索引 | 是 | 是 | 是 | 是 |
| 多索引鍵索引 | 是 | 是 | 是 | 是 |
| 文字索引 | 否 | 否 | 是 | 否 |
| 2dsphere | 是 | 是 | 是 | 否 |

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| 2d 索引 | 否 | 否 | 否 | 否 |
| 雜湊索引 | 否 | 否 | 否 | 否 |

索引屬性

| Command | 3.6 | 4.0 | 5.0 | 彈性叢集 |
|---------|-----|-----|-----|------|
| TTL | 是 | 是 | 是 | 是 |
| 唯一 | 是 | 是 | 是 | 是 |
| 部分 | 否 | 否 | 是 | 否 |
| 不區分大小寫 | 否 | 否 | 否 | 否 |
| 稀疏 | 是 | 是 | 是 | 是 |
| 背景介紹 | 是 | 是 | 是 | 否 |

Amazon DocumentDB 生成人工智能

Amazon DocumentDB 提供的功能可讓機器學習 (ML) 和生成人工智慧 (AI) 模型即時處理儲存在 Amazon DocumentDB 中的資料。客戶不再需要花費時間管理個別的基礎結構、撰寫程式碼與其他服務連線，以及從其主要資料庫複製資料。

如需有關人工智慧 AWS 以及如何支援 AI 需求的詳細資訊，請參閱這篇 [「What is」](#) 文章。

主題

- [使用 Amazon SageMaker 畫布進行無程式碼機器](#)
- [矢量搜索 Amazon DocumentDB](#)

使用 Amazon SageMaker 畫布進行無程式碼機器

[Amazon SageMaker Canvas](#) 可讓您建立自己的 AI/ML 模型，而不需要撰寫一行程式碼。您可以針對常見使用案例 (例如回歸和預測) 建立機器學習模型，並從 Amazon Bedck 存取和評估基礎模型 (FM)。您也可以從 Amazon SageMaker JumpStart 存取公有 FM，進行內容產生、文字擷取和文字摘要，以支援生成式 AI 解決方案。

如何使用 Canvas 構建無代碼 ML 模型 SageMaker

Amazon DocumentDB 現在與 Amazon SageMaker 畫布整合，以啟用儲存在 Amazon DocumentDB 中的數據的無代碼機器學習 (ML)。您現在可以針對回歸和預測需求建置 ML 模型，並使用儲存在 Amazon DocumentDB 中的資料，使用基礎模型進行內容摘要和產生，而無需撰寫單行程式碼。

SageMaker Canvas 提供視覺化介面，可讓 Amazon DocumentDB 客戶產生預測，而不需要任何 AI/ML 專業知識或撰寫單行程式碼。客戶現在可以從 Amazon DocumentDB 資料啟動 SageMaker 畫布工作區 AWS Management Console、匯入和加入資料準備和模型訓練。Amazon DocumentDB 中的資料現在可以在 SageMaker Canvas 中使用來建立和擴充模型，以預測客戶流失、偵測詐騙、預測維護失敗、預測商業指標以及產生內容。客戶現在可以使用 SageMaker Canvas 與 Amazon 的原生整合，跨團隊發佈和分享 ML 導向的洞察。QuickSight SageMaker Canvas 中的資料擷取管道預設會在 Amazon DocumentDB 次要執行個體上執行，以確保應用程式和 SageMaker Canvas 擷取工作負載的效能不會受到影響。

Amazon DocumentDB 客戶可以瀏覽至新的 Amazon DocumentDB 無程式碼 ML 主控台頁面，並連線至新的或可用的畫布工作區，開始使用 SageMaker Canvas。SageMaker

設定 SageMaker 網域和使用者設定檔

您可以從僅限 VPC 模式下執行的 SageMaker 網域連線到 Amazon DocumentDB 叢集。透過在 VPC 中啟動 SageMaker 網域，您可以從 SageMaker Studio 和 Canvas 環境控制資料流程。這可讓您限制網際網路存取、使用標準網 AWS 路和安全性功能監控和檢查流量，以及透過 VPC 端點連線至其他 AWS 資源。請參閱[亞馬遜 SageMaker 畫布入門](#)和在[沒有網際網路存取權的 VPC 中設定 Amazon SageMaker Canvas](#) 位於 Amazon 開 SageMaker 發人員指南中，以建立您的網 SageMaker 域以連接到 Amazon DocumentDB 叢集。

設定 Amazon DocumentDB 和畫布的 IAM 存取權限 SageMaker

已AmazonDocDBConsoleFullAccess附加至其關聯角色和身分的 Amazon DocumentDB 使用者可以存取. AWS Management Console將下列動作新增至上述角色或身分，以便透過 Amazon SageMaker Canvas 存取無程式碼機器學習。

```
"sagemaker:CreatePresignedDomainUrl",
"sagemaker:DescribeDomain",
"sagemaker:ListDomains",
"sagemaker:ListUserProfiles"
```

為 SageMaker Canvas 建立資料庫使用者和角色

您可以使用 Amazon DocumentDB 中的角色型存取控制 (RBAC) 來限制使用者可在資料庫上執行的動作的存取。RBAC 的運作方式是將一或多個角色授予使用者。這些角色決定使用者可以對資料庫資源執行的作業。

身為畫布使用者，您可以使用使用者名稱和密碼登入資料連線到 Amazon DocumentDB 資料庫。您可以為 Canvas 使用者建立資料庫使用者/角色，該使用者可以使用 Amazon 文件 DBB RBAC 功能對特定資料庫具有讀取存取權限。

例如，使用以下createUser操作：

```
db.createUser({
  user: "canvas_user",
  pwd: "<insert-password>",
  roles: [{role: "read", db: "sample-database-1"}]
})
```

這將創建一個 `canvas_user` 個具有對 `sample-database-1` 數據庫的讀取權限。您的畫布分析師可以使用此登入資料存取 Amazon DocumentDB 叢集中的資料。請參閱以瞭解 [使用角色型存取控制存取資料庫](#) 解更多資訊。

可用地區

無程式碼整合適用於同時支援 Amazon DocumentDB 和 Amazon SageMaker 畫布的區域。這些地區包括：

- us-east-1(維吉尼亞北部)
- us-east-2(俄亥俄)
- us-west-2(奧勒岡)
- ap-northeast-1(東京)
- ap-northeast-2(首爾)
- ap-south-1(孟買)
- ap-southeast-1(新加坡)
- ap-southeast-2(雪梨)
- eu-central-1(法蘭克福)
- eu-west-1(愛爾蘭)

請參閱 [Amazon 開 SageMaker 發人員指南中的 Amazon SageMaker Canvas](#)，了解最新的區域可用性。

向量搜索 Amazon DocumentDB

向量搜尋是機器學習中使用的一種方法，透過使用距離或相似度量來比較其向量表示法，藉此找出與指定資料點相似的資料點。向量空間中的兩個向量越接近，基礎項目被認為就越相似。這種技術有助於捕獲數據的語義含義。此方法適用於各種應用程式，例如推薦系統、自然語言處理和影像辨識。

Amazon DocumentDB 的向量搜尋結合 JSON 文件資料庫的彈性和豐富的查詢功能，以及向量搜尋的強大功能。如果您想要使用現有的 Amazon DocumentDB 資料或靈活的文件資料結構來建立機器學習和生成 AI 使用案例 (例如語義搜尋體驗、產品推薦、個人化、聊天機器人、詐騙偵測和異常偵測)，Amazon DocumentDB 向量搜尋是您的理想選擇。向量搜尋可在 Amazon DocumentDB 5.0 執行個體型叢集上使用。

主題

- [插入向量](#)
- [建立向量索引](#)
- [取得索引定義](#)
- [查詢向量](#)
- [功能和限制](#)
- [最佳實務](#)

插入向量

若要將向量插入 Amazon DocumentDB 資料庫，您可以使用現有的插入方法：

範例

在下面的例子中，一個測試數據庫中的五個文檔的集合被創建。每個文件都包含兩個欄位：產品名稱及其對應的向量嵌入。

```
db.collection.insertMany([
  {"product_name": "Product A", "vectorEmbedding": [0.2, 0.5, 0.8]},
  {"product_name": "Product B", "vectorEmbedding": [0.7, 0.3, 0.9]},
  {"product_name": "Product C", "vectorEmbedding": [0.1, 0.2, 0.5]},
  {"product_name": "Product D", "vectorEmbedding": [0.9, 0.6, 0.4]},
  {"product_name": "Product E", "vectorEmbedding": [0.4, 0.7, 0.2]}
]);
```

建立向量索引

Amazon DocumentDB 同時支援階層式可導覽小型世界 (HNSW) 索引和具有扁平壓縮的反轉檔案 (IVFFLAT) 索引方法。IVFFlat 索引會將向量分隔為清單，然後搜尋最接近查詢向量的清單中選取的子集。另一方面，HNSW 索引會將向量資料組織成多層圖形。雖然 HNSW 的構建時間比 IVFFlat 慢，但它提供了更好的查詢性能和調用。與 IVFFlat 不同，HNSW 不涉及任何訓練步驟，允許在沒有任何初始數據負載的情況下生成索引。對於大多數使用案例，我們建議使用 HNSW 索引類型進行向量搜尋。

如果您不建立向量索引，Amazon DocumentDB 會執行完全最近的鄰近搜尋，以確保完美回收。但是，在生產場景中，速度至關重要。我們建議使用向量索引，這可能會交易一些回收以提高速度。重要的是要注意，添加向量索引可能會導致不同的查詢結果。

範本

您可以使用以下 `createIndex` 或 `runCommand` 模板在向量字段上構建向量索引：

Using createIndex

在某些驅動程式 (例如 mongosh 和 Java) 中，使用中的vectorOptions參數createIndex可能會導致錯誤。在這種情況下，我們建議使用runCommand：

```
db.collection.createIndex(  
  { "<vectorField>": "vector" },  
  { "name": "<indexName>",  
    "vectorOptions": {  
      "type": " <hnsw> | <ivfflat> ",  
      "dimensions": <number_of_dimensions>,  
      "similarity": " <euclidean> | <cosine> | <dotProduct> ",  
      "lists": <number_of_lists> [applicable for IVFFlat],  
      "m": <max number of connections> [applicable for HNSW],  
      "efConstruction": <size of the dynamic list for index build> [applicable for  
HNSW]  
    }  
  }  
);
```

Using runCommand

在某些驅動程式 (例如 mongosh 和 Java) 中，使用中的vectorOptions參數createIndex可能會導致錯誤。在這種情況下，我們建議使用runCommand：

```
db.runCommand(  
  { "createIndexes": "<collection>",  
    "indexes": [{  
      key: { "<vectorField>": "vector" },  
      vectorOptions: {  
        type: " <hnsw> | <ivfflat> ",  
        dimensions: <number of dimensions>,  
        similarity: " <euclidean> | <cosine> | <dotProduct> ",  
        lists: <number_of_lists> [applicable for IVFFlat],  
        m: <max number of connections> [applicable for HNSW],  
        efConstruction: <size of the dynamic list for index build> [applicable for  
HNSW]  
      },  
      name: "myIndex"  
    }]  
  }  
);
```

| 參數 | 需求 | 資料類型 | 描述 | 數值 |
|-----------------------|--------|---------|--|--|
| name | 選擇性 | string | 指定索引的名稱。 | 英數字元 |
| type | 選擇性 | | 指定索引的類型。 | 支持：高壓或常見 預設值：HNSW (引擎修補程式 3.0.4574 之後) |
| dimensions | 必要 | integer | 指定向量資料中的維度數。 | 最多可容納 2,000 個維度。 |
| similarity | 必要 | string | 指定用於相似度計算的距離公制。 | <ul style="list-style-type: none"> • euclidean • cosine • dotProduct |
| lists | 需要為伊夫拉 | integer | 指定 IVFFlat 索引用來群組向量資料的叢集數目。建議的設定是最多 1M 文件和 $\sqrt{\text{# of documents}}$ 超過 1M 文件的文件數量 /1000。 | 下限：1 最大值：請參閱 下功能和限制 方每個執行個體類型表格的清單。 |
| m | 選擇性 | integer | 指定 HNSW 索引的最大連接數 | 預設：16 範圍 [2, 100] |
| efConstruction | 選擇性 | integer | 指定用於建構 HNSW 索引圖形的動態候選清單大小。 | 預設：64 範圍 [4, 千] |

| 參數 | 需求 | 資料類型 | 描述 | 數值 |
|----|----|------|-------------------------------|----|
| | | | efConstruction 必須大於或等於 (2* m) | |

重要的是要適當地設置子參數的值，lists 例如 IVFFlat m 和 HNSW，因為這將影響搜索的準確性/調用 efConstruction，構建時間和性能。較高的清單值會增加查詢的速度，因為它會減少每個清單中的向量數目，從而產生較小的區域。但是，較小的區域大小可能會導致更多的召回錯誤，從而導致較低的精度。對於 HNSW，增加的值 m 並 efConstruction 增加準確度，但也會增加索引建立時間和大小。請參閱以下範例：

範例

HNSW

```
db.collection.createIndex(
  { "vectorEmbedding": "vector" },
  { "name": "myIndex",
    "vectorOptions": {
      "type": "hnsw",
      "dimensions": 3,
      "similarity": "euclidean",
      "m": 16,
      "efConstruction": 64
    }
  }
);
```

IVFFlat

```
db.collection.createIndex(
  { "vectorEmbedding": "vector" },
  { "name": "myIndex",
    "vectorOptions": {
      "type": "ivfflat",
      "dimensions": 3,
      "similarity": "euclidean",
      "lists": 1
    }
  }
);
```

```
)
```

取得索引定義

您可以使用以下`getIndexes`命令檢視索引的詳細資訊，包括向量索引：

範例

```
db.collection.getIndexes()
```

範例輸出

```
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "test.collection"
  },
  {
    "v" : 4,
    "key" : {
      "vectorEmbedding" : "vector"
    },
    "name" : "myIndex",
    "vectorOptions" : {
      "type" : "ivfflat",
      "dimensions" : 3,
      "similarity" : "euclidean",
      "lists" : 1
    },
    "ns" : "test.collection"
  }
]
```

查詢向量

向量查詢範本

使用下列範本查詢向量：

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": <query vector>,
        "path": "<vectorField>",
        "similarity": "<distance metric>",
        "k": <number of results>,
        "probes":<number of probes> [applicable for IVFFlat],
        "efSearch":<size of the dynamic list during search> [applicable for HNSW]
      }
    }
  }
]);
```

| 參數 | 需求 | Type | 描述 | 數值 |
|---------------------|-----|----------|-----------------------|--|
| vectorSearch | 必要 | operator | 裡面使用 \$ 搜索命令來查詢向量。 | |
| vector | 必要 | 陣列 | 指出將用來尋找類似向量的查詢向量。 | |
| path | 必要 | string | 定義向量欄位的名稱。 | |
| k | 必要 | integer | 指定搜尋傳回的結果數目。 | |
| similarity | 必要 | string | 指定用於相似度計算的距離公式。 | <ul style="list-style-type: none"> • euclidean • cosine • dotProduct |
| probes | 選擇性 | integer | 要向量搜尋檢查的叢集數目。較高的值以速度代 | 預設：1 |

| 參數 | 需求 | Type | 描述 | 數值 |
|-----------------|-----|---------|--|--------------------|
| | | | 價提供更好的召回效果。它可以設置為列表的數量，以進行確切的最近鄰搜索（此時計劃器將不會使用索引）。開始微調的建議設定為。sqrt(# of lists) | |
| efSearch | 選擇性 | integer | 指定 HNSW 索引在搜尋期間使用的動態候選清單的大小。較高的值以速度代價 efSearch 提供更好的召回。 | 預設：40 範圍 [1, 千] |

請務必微調 efSearch (HNSW) 或 probes (IVFlat) 的值，以達到所需的效能和準確度。請參閱下列範例作業：

HNSW

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": [0.2, 0.5, 0.8],
        "path": "vectorEmbedding",
        "similarity": "euclidean",
        "k": 2,
        "efSearch": 40
      }
    }
  }
])
```

```
]);
```

IVFFlat

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": [0.2, 0.5, 0.8],
        "path": "vectorEmbedding",
        "similarity": "euclidean",
        "k": 2,
        "probes": 1
      }
    }
  }
]);
```

範例輸出

此操作的輸出如下所示：

```
{ "_id" : ObjectId("653d835ff96bee02cad7323c"), "product_name" : "Product A",
  "vectorEmbedding" : [ 0.2, 0.5, 0.8 ] }
{ "_id" : ObjectId("653d835ff96bee02cad7323e"), "product_name" : "Product C",
  "vectorEmbedding" : [ 0.1, 0.2, 0.5 ] }
```

功能和限制

版本相容性

- Amazon DocumentDB 的向量搜索僅適用於基於 Amazon DocumentDB 5.0 實例的集群。

向量

- Amazon DocumentDB 可以索引多達 2,000 個維度的向量。但是，最多可以在沒有索引的情況下存儲 16,000 個維度。

索引

- 對於創建 IVFFlat 索引，列表參數的建議設置是最多 1M 文檔和超過 1M 文檔的文檔/1000 $\sqrt{\text{of documents}}$ 的數量。由於工作記憶體限制，Amazon DocumentDB 支援清單參數的特定最大值，具體取決於維度數目。下表提供了 500、1000 和 2,000 維向量的列表參數的最大值，供您參考：

| 執行個體類型 | 具有 500 個維度的清單 | 具有 1000 個維度的清單 | 具有 2000 個維度的清單 |
|---------|---------------|----------------|----------------|
| t3. | 372 | 257 | 150 |
| R5.l | 915 | 741 | 511 |
| R5.xl | 1,393 | 1,196 | 901 |
| 紅 | 5,460 | 5,230 | 4,788 |
| r5.4xl | 7,842 | 7,599 | 7,138 |
| r5.8xl | 11,220 | 10,974 | 10,498 |
| r5.12xl | 13,774 | 13,526 | 13,044 |
| r5.16xl | 15,943 | 15,694 | 15,208 |
| r5.24xl | 19,585 | 19,335 | 18,845 |

- 向量索引不支援其他索引選項 compound，例如 sparse 或 partial。
- HNSW 索引不支援並行索引建立。它僅支持 IVFLAT 指數。

向量查詢

- 對於向量搜尋查詢，請務必微調參數，例如 probes 或以 efSearch 取得最佳結果。probes 或 efSearch 參數的值越高，回收越高，速度也越低。開始微調探棒參數的建議設定為 $\sqrt{\text{of lists}}$ 。

最佳實務

了解在 Amazon DocumentDB 中使用向量搜尋的最佳實務。在新最佳實務確定時，會不斷更新此小節。

- 具有扁平壓縮 (IVFFLAT) 索引創建的反向文件涉及聚類和組織基於相似之處的數據點。因此，為了使索引更有效，我們建議您在創建索引之前至少加載一些數據。
- 對於向量搜尋查詢，請務必微調參數，例如probes或以efSearch取得最佳結果。probes或efSearch參數的值越高，回收越高，速度越低。開始微調probes參數的建議設置為sqrt(lists)。

資源

- [向量搜索什麼是新的博客文章](#)
- [語意搜尋程式碼範例](#)
- [Amazon DocumentDB 向量搜索代碼示例](#)

遷移到 Amazon DocumentDB

Amazon DocumentDB 資料庫 (與 MongoDB 相容性) 是一種與 MongoDB API 相容的全受管資料庫服務。您可以使用本節詳述的程序，將資料從內部部署或 Amazon 彈性運算雲端 (Amazon EC2) 上執行的 MongoDB 資料庫遷移到 Amazon DocumentDB。

主題

- [使用升級您的 Amazon DocumentDB 集群 AWS Database Migration Service](#)
- [遷移工具](#)
- [探索](#)
- [規劃：Amazon DocumentDB 叢集需求](#)
- [遷移方法](#)
- [遷移來源](#)
- [遷移連線能力](#)
- [測試](#)
- [效能測試](#)
- [容錯移轉測試](#)
- [其他資源](#)
- [遷移教戰手冊：MongoDB 到 Amazon DocumentDB](#)

使用升級您的 Amazon DocumentDB 集群 AWS Database Migration Service

Important

Amazon DocumentDB 不遵循相同的支持生命週期作為 MongoDB 和 MongoDB 的 end-of-life 計劃不適用於 Amazon DocumentDB。目前沒有針 end-of-life 對 Amazon DocumentDB 3.6 的計劃，您現有的 MongoDB 3.6 驅動程序，應用程序和工具將繼續與 Amazon DocumentDB 一起使用。

您可以使用將 Amazon DocumentDB 叢集升級到更高版本，並將停機時間降至最低。AWS DMS 這項全受管服務，可讓您輕鬆地從舊版 Amazon DocumentDB 版本、關聯式資料庫和非關聯式資料庫遷移到目標 Amazon DocumentDB 叢集。

主題

- [步驟 1：啟用變更串流](#)
- [步驟 2：修改變更串流保留持續時間](#)
- [步驟 3：遷移索引](#)
- [步驟 4：建立 AWS DMS 複製執行個體](#)
- [步驟 5：建立 AWS DMS 來源端點](#)
- [步驟 6：建立 AWS DMS 目標端點](#)
- [步驟 7：建立並執行移轉任務](#)
- [步驟 8：將應用程式端點變更為目標 Amazon DocumentDB 叢集](#)

步驟 1：啟用變更串流

若要執行最短停機時間移轉，AWS DMS 需要存取叢集的變更串流。[Amazon DocumentDB 變更串流](#)可提供叢集集合和資料庫中發生的更新事件，依時間順序排列。從變更串流讀取可 AWS DMS 以執行變更資料擷取 (CDC)，並將增量更新套用至目標 Amazon DocumentDB 叢集。

若要為特定資料庫上的所有集合啟用變更串流，請使用 mongo 殼層對 Amazon DocumentDB 叢集進行驗證，然後執行下列命令：

```
db.adminCommand({modifyChangeStreams: 1,
  database: "db_name",
  collection: "",
  enable: true});
```

步驟 2：修改變更串流保留持續時間

接下來，根據您希望在變更串流中保留變更事件的時間長度來修改變更串流保留期間。例如，如果您預期 Amazon DocumentDB 叢集遷移需 AWS DMS 要 12 小時的時間，則應將變更串流保留設定為大於 12 小時的值。Amazon DocumentDB 叢集的預設保留期為三小時。您可以使用 AWS Management Console 或將 Amazon DocumentDB 叢集的變更串流日誌保留持續時間修改為一小時到七天之間。AWS CLI 如需詳細資訊，請參閱[修改變更串流記錄保留持續時間](#)。

步驟 3：遷移索引

在來源 Amazon DocumentDB 叢集上的目標 Amazon 文件資料庫叢集上建立相同的索引。雖然會 AWS DMS 處理資料的移轉，但不會移轉索引。若要移轉索引，請使用 Amazon 文件資料庫索引工具從來源 Amazon DocumentDB 叢集匯出索引。您可以建立 Amazon DocumentDB 工具 GitHub 存放庫的複製，並遵循中的指示來取得此工具。[README.md](#)您可以從 Amazon EC2 執行個體或在與 Amazon DocumentDB 叢集相同的 Amazon VPC 中執行的 AWS Cloud9 環境執行此工具。

在下列範例中，將每個#####取代為您自己的資訊。

下列程式碼會從您的來源 Amazon DocumentDB 叢集傾印索引：

```
python migrationtools/documentdb_index_tool.py --dump-indexes
--uri mongodb://sample-user:user-password@sample-source-cluster.node.us-east-1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
--dir ~/index.js/

2020-02-11 21:51:23,245: Successfully authenticated to database: admin2020-02-11
21:46:50,432: Successfully connected to instance docdb-40-xx.cluster-xxxxxxx.us-east-1.docdb.amazonaws.com:27017
2020-02-11 21:46:50,432: Retrieving indexes from server...2020-02-11 21:46:50,440:
Completed writing index metadata to local folder: /home/ec2-user/index.js/
```

成功匯出索引之後，請在目標 Amazon DocumentDB 叢集中還原這些索引。若要還原您在上一個步驟中匯出的索引，請使用 Amazon DocumentDB 索引工具。下列命令會從指定的目錄還原目標 Amazon DocumentDB 叢集中的索引。

```
python migrationtools/documentdb_index_tool.py --restore-indexes
--uri mongodb://sample-user:user-password@sample-destination-cluster.node.us-east-1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
--dir ~/index.js/

2020-02-11 21:51:23,245: Successfully authenticated to database: admin2020-02-11
21:51:23,245: Successfully connected to instance docdb-50-xx.cluster-xxxxxxx.us-east-1.docdb.amazonaws.com:27017
2020-02-11 21:51:23,264: testdb.coll: added index: _id
```

若要確認您已正確還原索引，請使用 mongo 殼層連接到目標 Amazon DocumentDB 叢集，並列出指定集合的索引。請參閱以下代碼：


```
mongo --ssl
--host docdb-xx-xx.cluster-xxxxxxx.us-east-1.docdb.amazonaws.com:27017
--sslCAFile rds-ca-2019-root.pem --username documentdb --password documentdb

db.coll.getIndexes()
```

步驟 4：建立 AWS DMS 複製執行個體

AWS DMS 複製執行個體會從您的來源 Amazon DocumentDB 叢集連接和讀取資料，並將其寫入目標 Amazon DocumentDB 叢集。AWS DMS 複製執行個體可以執行大量載入和 CDC 作業。大部分這種處理都發生在內存中。但是，大型操作可能需要在磁盤上進行一些緩衝。快取交易和日誌檔案也會寫入磁碟。移轉資料之後，複製執行個體也會串流任何變更事件，以確保來源和目標同步。

若要建立 AWS DMS 複製執行個體：

1. 開啟主 AWS DMS [控制台](#)。
2. 在導覽窗格中，選擇 Replication instances (複製執行個體)。
3. 選擇 Create replication instance (建立複製執行個體) 並輸入下列資訊：
 - 在「名稱」中，輸入您選擇的名稱。例如 docdb36todocdb40。
 - 在說明中，輸入您選擇的描述。對於列表，Amazon DocumentDB 3.6 到 Amazon DocumentDB 4.0 複製實例。
 - 對於實例類，請根據需要選擇大小。
 - 對於引擎版本，請選擇 3.4.1。
 - 對於 Amazon 虛擬私人雲端，請選擇用來存放您的來源和目標 Amazon DocumentDB 叢集的亞馬遜 VPC。
 - 對於已配置的儲存區 (GiB)，請使用預設值 50 GiB。如果您的寫入輸送量工作負載很高，請增加此值以符合您的工作負載。
 - 對於異地同步備份，如果您需要高可用性和容錯移轉支援，請選擇是。
 - 對於 Publicly accessible (可公開存取)，啟用此選項。

Replication instance configuration

Name
The name must be unique among all of your replication instances in the current AWS region.

Replication instance name must not start with a numeric value

Description

The description must only have unicode letters, digits, whitespace, or one of these symbols: _:/=+-.@. 1000 maximum character.

Instance class [Info](#)
Choose an appropriate instance class for your replication needs. Each instance class provides differing levels of compute, network and memory capacity. [DMS pricing](#)

16 vCPUs 30 GiB Memory

Include previous-generation instance classes

Engine version
Choose an AWS DMS version to run on your replication instance. [DMS versions](#)

Include Beta DMS versions

Allocated storage (GiB)
Choose the amount of storage space you want for your replication instance. AWS DMS uses this storage for log files and cached transactions while replication tasks are in progress.

VPC
Choose an Amazon Virtual Private Cloud (VPC) where your replication instance should run.

Multi AZ
If you choose this option, AWS DMS will perform a multi-AZ deployment, with a primary instance in one availability zone (AZ) and a standby instance in another AZ. This configuration provides a highly available, fault-tolerant replication environment. Billing is based on [DMS pricing](#)

Publicly accessible
If you choose this option, AWS DMS will assign a public IP address to your replication instance, and you'll be able to connect to databases outside of your Amazon VPC.

4. 選擇 Create replication instance (建立複寫執行個體)。

步驟 5：建立 AWS DMS 來源端點

來源端點用於來源 Amazon DocumentDB 叢集。

建立來源端點

1. 開啟主 AWS DMS [控制台](#)。
2. 在導覽窗格中選擇端點。
3. 選擇 Create endpoint 並輸入下列資訊：
 - 對於 Endpoint type (端點類型)，選擇 Source (來源)。
 - > 對於端點識別碼，請輸入容易記住的名稱，例如 docdb-source。
 - 針對「來源引擎」，選擇 docdb。
 - 針對伺服器名稱，輸入來源 Amazon DocumentDB 叢集的 DNS 名稱。
 - 在連接埠中，輸入來源 Amazon DocumentDB 叢集的連接埠號碼。
 - 對於 SSL 模式，請選擇 verify-full。
 - 對於 CA 憑證，請選擇 [新增 CA 憑證]。下載 [新的 CA 憑證](#) 以建立 TLS 連線服務包。在「憑證識別碼」中，輸入 rds-combined-ca-bundle。對於 Import certificate file (匯入憑證檔案)，請選擇 Choose file (選擇檔案)，然後瀏覽至您先前下載的 .pem 檔案。選取並開啟檔案。選擇匯入憑證，然後 rds-combined-ca-bundle 從選擇憑證下拉式清單中選擇
 - 在使用者名稱中，輸入來源 Amazon DocumentDB 叢集的主要使用者名稱。
 - 對於密碼，請輸入來源 Amazon DocumentDB 叢集的主要密碼。
 - 在「資料庫名稱」中，輸入您要升級的資料庫名稱。

Endpoint configuration

Endpoint identifier [Info](#)
A label for the endpoint to help you identify it.

Source engine
The type of database engine this endpoint is connected to.

Server name

Port
The port the database runs on for this endpoint.

Secure Socket Layer (SSL) mode
The type of Secure Socket Layer enforcement

CA certificate

 [Add new CA certificate](#)

User name [Info](#)

Password [Info](#)

Database name

4. 測試您的連線以確認是否已成功設定。

▼ **Test endpoint connection (optional)**

VPC
vpc-2bf12540

Replication instance
A replication instance performs the database migration
docdb36todocdb40

Run test

| Endpoint identifier | Replication instance | Status | Message |
|---------------------|----------------------|------------|---------|
| docdb36-source | docdb36todocdb40 | successful | |

5. 選擇建立端點。

Note

AWS DMS 一次只能移轉一個資料庫。

步驟 6：建立 AWS DMS 目標端點

目標端點適用於您的目標 Amazon DocumentDB 叢集。

若要建立目標端點：

1. 開啟 [AWS DMS 主控台](#)。
2. 在導覽窗格中選擇端點。
3. 選擇 Create endpoint (建立端點) 並輸入下列資訊：
 - 對於 Endpoint type (端點類型)，選擇 Target (目標)。
 - 對於 Endpoint identifier (端點識別符)，輸入易記的名稱，例如 docdb-target。
 - 針對「來源引擎」，選擇 docdb。
 - 針對伺服器名稱，輸入目標 Amazon DocumentDB 叢集的 DNS 名稱。
 - 在連接埠中，輸入目標 Amazon DocumentDB 叢集的連接埠號碼。

- 對於 SSL 模式，請選擇 `verify-full`。
- 若為 CA 憑證，請從 [選擇 `rds-combined-ca-bundle` 憑證] 下拉式清單中選擇現有憑證。
- 在使用者名稱中，輸入目標 Amazon DocumentDB 叢集的主要使用者名稱。
- 對於密碼，請輸入目標 Amazon DocumentDB 叢集的主要密碼。
- 在「資料庫名稱」中，輸入您用來設定來源端點的相同資料庫名稱。

Endpoint configuration

Endpoint identifier [Info](#)
A label for the endpoint to help you identify it.

Target engine
The type of database engine this endpoint is connected to.
Server name

Port
The port the database runs on for this endpoint.

Secure Socket Layer (SSL) mode
The type of Secure Socket Layer enforcement.

CA certificate
 [Add new CA certificate](#)
User name [Info](#)

Password [Info](#)

Database name

4. 測試您的連線以確認已成功設定。

▼ Test endpoint connection (optional)

VPC
vpc-2bf12540

Replication instance
A replication instance performs the database migration
docdb36todocdb40

Run test

| Endpoint identifier | Replication instance | Status | Message |
|---------------------|----------------------|------------|---------|
| docdb36-target | docdb36todocdb40 | successful | |

5. 選擇建立端點。

步驟 7：建立並執行移轉任務

AWS DMS 工作會將複製執行個體與來源和目標執行個體繫結。建立移轉工作時，您可以指定來源端點、目標端點、複製執行個體以及任何所需的移轉設定。您可以使用三種不同的移轉類型來建立 AWS DMS 工作：移轉現有資料、移轉現有資料，以及複製進行中的變更或僅複製資料變更。由於此逐步解說的目的是以最短的停機時間升級 Amazon DocumentDB 叢集，因此這些步驟利用遷移現有資料和複製進行中的變更的選項。使用此選項，可在移轉現有資料時 AWS DMS 擷取變更。AWS DMS 即使載入大量資料後，仍會繼續擷取和套用變更。來源和目標資料庫最終將保持同步，最大限度地減少停機時間。

以下是針對最短停機時間移轉建立移轉任務的步驟：

1. 開啟主 AWS DMS [控制台](#)。
2. 在導覽窗格中，選擇 Tasks (任務)。
3. 選擇 Create task (建立任務) 並輸入下列資訊：
 - 對於 Task name (任務名稱)，輸入易記的名稱，例如 my-dms-upgrade-task。
 - 針對複製執行個體，選擇您在 [步驟 3：建立複製執行個體中建立的 AWS Database Migration Service 複製執行個體](#)

- 針對來源端點，選擇您在[步驟 4：建立來源端點中建立的 AWS Database Migration Service 來源端點](#)
- 針對目標端點，選擇您在[步驟 5：建立目標端點中建立的 AWS Database Migration Service 目標端點](#)
- 對於移轉類型，請選擇移轉現有資料並複寫進行中的變更。

Task configuration

Task identifier

Replication instance

Source database endpoint

Target database endpoint

Migration type [Info](#)

4. 在「工作設定」區段中，啟用 CloudWatch 記錄檔。
5. 在「表格對應」段落中，選擇不執行任何動作。這將確保在步驟 3 中創建的索引不會被刪除。
6. 對於「移轉」工作啟動組態，請選擇「建立時自動」。這會在您建立移轉任務後自動啟動它。
7. 選擇 Create task (建立任務)。

AWS DMS 現在開始將資料從您的來源 Amazon 文件資料庫叢集遷移到目標 Amazon DocumentDB 叢集。工作狀態應該從 [開始] 變更為 [執行中]。您可以選擇 AWS DMS 主控台中的 [工作] 來監視進度。幾分鐘/小時後 (視您的移轉大小而定)，狀態應從 [載入完成] 變更為 [載入完成]，複寫正在進行中。這表示 AWS DMS 已完成將來源 Amazon DocumentDB 叢集的完整負載移轉到目標 Amazon DocumentDB 叢集，現在正在複寫變更事件。

| Summary | | | |
|--------------------------------------|--------------------------------|---------------|---------------|
| Status | Type | Source | Target |
| 🟢 Load complete, replication ongoing | Full load, ongoing replication | docdb36source | docdb40target |

最終您的源和目標將保持同步。您可以透過對集合執行 `count()` 作業來確認它們是否處於同步狀態，以確認所有變更事件都已移轉。

步驟 8：將應用程式端點變更為目標 Amazon DocumentDB 叢集

完成滿載且 CDC 程序持續複寫之後，您就可以準備好將應用程式的資料庫連線端點從來源 Amazon DocumentDB 叢集變更為目標 Amazon DocumentDB 叢集。

遷移工具

若要移轉至 Amazon DocumentDB，大多數客戶使用的兩個主要工具是 [AWS Database Migration Service \(AWS DMS\)](#) 和命令列公用程式，例如 `mongodump` 和 `mongoexport`。最佳做法是對於上述任一選項，建議您在開始遷移之前先在 Amazon DocumentDB 中建立索引，因為這樣可以縮短整體時間並提高遷移速度。要做到這一點，您可以使用 [Amazon DocumentDB 索引](#) 工具。

AWS Database Migration Service

AWS Database Migration Service (AWS DMS) 是一種雲端服務，可讓您輕鬆將關聯式資料庫和非關聯式資料庫遷移到 Amazon DocumentDB。您可以使用 AWS DMS 將資料從現場部署或 EC2 上託管的資料庫遷移到 Amazon DocumentDB。使用 AWS DMS，您可以執行一次性移轉，或複寫進行中的變更，以使來源和目標保持同步。

如需使用移轉 AWS DMS 至亞馬遜文件資料庫的詳細資訊，請參閱：

- [使用 MongoDB 作為一個源代碼 AWS DMS](#)
- [使用 Amazon DocumentDB 作為目標 AWS Database Migration Service](#)
- [逐步解說：從 MongoDB 遷移到 Amazon DocumentDB](#)

命令列公用程式

在 Amazon DocumentDB 間移轉資料的常見公用程式包括 `mongodump`、`mongoexport`、`mongoimport`、`mongorestore` 和 `mongoexport`。一般是最有效的公用程式，因為它們會以二進位格式從資料庫傾印和還原資料。與邏輯匯出相比，這通常是效能最佳的選項，產生的資料大小較小。如果您想以 JSON 或 CSV 等邏輯格式匯

出和匯入資料，`mongoexport` 和 `mongoimport` 很實用，因為人類可讀取這些資料，但通常比 `mongodump/mongorestore` 慢並且會產生更大的資料大小。

下面的[遷移方法](#)部分將根據您的用例 AWS DMS 和需求討論何時最好使用和命令行實用程序。

探索

針對每個 MongoDB 部署，您應該識別和記錄兩組資料：「架構詳細資訊」和「營運特點」。此資訊可協助您選擇適當的遷移方法和叢集大小。

架構詳細資訊

- 名稱

選擇追蹤此部署的唯一名稱。

- 版本

記錄您部署正在執行的 MongoDB 版本。若要尋找版本，請連線到具有 `mongo` 殼層的複本集成員，執行 `db.version()` 操作。

- 類型

記錄您的部署為獨立的 `mongo` 執行個體、複本集，還是碎片叢集。

- 成員

記錄每個叢集的主機名稱、地址和連接埠，以及複本集或獨立成員。

針對叢集部署，您可以透過連線到具有 `mongo` 殼層的 `mongo` 主機，執行 `sh.status()` 操作，找到碎片成員。

針對複本集，您可以透過連線到具有 `mongo` 殼層的複本集成員，執行 `rs.status()` 操作，取得這些成員。

- Olog sizes (Oplog 大小)

針對複本集或碎片叢集，記錄每個複本集成員的 oplog 大小。若要尋找成員的 oplog 大小，請連線到具有 mongo 殼層的複本集成員，執行 `ps.printReplicationInfo()` 操作。

- Replica set member priorities (複本集成員優先順序)

針對複本集或碎片叢集，記錄每個複本集成員的優先順序。若要尋找複本集成員的優先順序，請連線到具有 mongo 殼層的複本集成員，執行 `rs.conf()` 操作。優先順序會顯示為 `priority` 金鑰值。

- TLS/SSL usage (TLS/SSL 用量)

記錄每個節點是否使用 Transport Layer Security (TLS)/Secure Sockets Layer (SSL) 在傳輸中加密。

營運特點

- Database statistics (資料庫統計資料)

針對每個集合，記錄以下資訊：

- 名稱
- 資料大小
- 集合計數

若要尋找資料庫統計資料，請連線到具有 mongo 殼層的資料庫，執行 `db.runCommand({dbstats: 1})` 命令。

- Collection statistics (集合統計資料)

針對每個集合，記錄以下資訊：

- 命名空間

- 資料大小
 - 索引計數
 - 集合是否受限
-
- Index statistics (索引統計資料)

針對每個集合，記錄以下索引資訊：

- 命名空間
- ID
- 大小
- 鍵
- TTL
- 稀疏
- 背景介紹

若要尋找索引資訊，請連線到具有 mongo 殼層的資料庫，執行 `db.collection.getIndexes()` 命令。

- Opcounters

此資訊有助於您了解目前的 MongoDB 工作負載模式 (大量讀取、大量寫入，或平衡)。同時也提供有關您初始 Amazon DocumentDB 執行個體選擇的指導。

以下是要在監控期間收集的幾項重要資訊 (計數/秒)：

- 查詢
- 插入
- 更新
- 刪除

您可以建立 `db.serverStatus()` 命令一段時間後的輸出圖表，取得此資訊。您也可以使用 `mongostat` 工具取得這些統計資料的即時值。但是，使用此選項要承擔在尖峰負載以外之用量期間規劃遷移的風險。

- Network statistics (網路統計資料)

此資訊有助於您了解目前的 MongoDB 工作負載模式 (大量讀取、大量寫入，或平衡)。同時也提供有關您初始 Amazon DocumentDB 執行個體選擇的指導。

以下是要在監控期間收集的幾項重要資訊 (計數/秒)：

- 連線
- 網路位元組輸入
- 網路位元組輸出

您可以建立 `db.serverStatus()` 命令一段時間後的輸出圖表，取得此資訊。您也可以使用 `mongostat` 工具取得這些統計資料的即時值。但是，使用此選項要承擔在尖峰負載以外之用量期間規劃遷移的風險。

規劃：Amazon DocumentDB 叢集需求

成功移轉需要仔細考慮 Amazon DocumentDB 叢集的組態，以及應用程式存取叢集的方式。決定叢集要求時請考慮下列各方面：

- 可用性

Amazon DocumentDB 透過複本執行個體的部署提供高可用性，這些複本執行個體可在稱為容錯移轉的程序中提升為主執行個體。將複本執行個體部署到不同的可用區域，您可以達到更高層級的可用性。

下表提供 Amazon DocumentDB 部署組態的準則，以符合特定的可用性目標。

| 可用性目標 | 執行個體總數 | 複本 | 可用區域 |
|--------|--------|----|------|
| 99% | 1 | 0 | 1 |
| 99.9% | 2 | 1 | 2 |
| 99.99% | 3 | 2 | 3 |

整體系統可靠性必須考慮所有元件，不只考慮資料庫。如需滿足整體系統可靠性需求的最佳實務與建議，請參閱 [AWS Well-Architected 的可靠性支柱](#) 白皮書。

- 效能

Amazon DocumentDB 執行個體可讓您讀取和寫入叢集的儲存磁碟區。叢集執行個體有多種類型，使用各種不同數量的記憶體和 vCPU，會影響您叢集的讀取和寫入效能。使用您在探索階段收集的資訊，選擇可支援您工作負載效能需求的執行個體類型。如需支援的執行個體類型清單，請參閱 [管理執行個體類別](#)。

為 Amazon DocumentDB 叢集選擇執行個體類型時，請考慮以下工作負載效能需求的方面：

- vCPUs：需要更高連線計數的架構可能會受益於具有更多 vCPUS 的執行個體。
- 記憶體 — 在可能的情況下，將工作資料集保留在記憶體中可提供最大效 起始準則是為 Amazon DocumentDB 引擎保留三分之一的執行個體記憶體，留下三分之二供工作資料集使用。
- 連線 — 每個 Amazon DocumentDB 執行個體 vCPU 的最低最佳連線計數為八個連線。雖然 Amazon DocumentDB 執行個體連線限制要高得多，但額外連線的效能優勢卻下降到每個 vCPU 的八個連線以上。

- 網路 — 具有大量用戶端或連線的工作負載應考量插入和擷取資料所需的彙總網路效能。大批量操作能更有效率地使用網路資源。
- 插入效能 — 單一文件插入通常是將資料插入 Amazon DocumentDB 中最慢的方式。大批量插入操作遠較單一插入更快。
- 讀取效能：從工作記憶體讀取的速度永遠比從儲存磁碟區傳回的讀取速度快。因此，最佳化執行個體記憶體大小以保留記憶體中的工作集，是最理想的方式。

除了從主執行個體提供讀取服務之外，Amazon DocumentDB 叢集也會自動設定為複本集。然後，您可以在 MongoDB 驅動程式中設定讀取偏好，路由唯讀查詢以讀取複本。您可以降低主執行個體的整體負載，透過新增複本來擴展讀取流量。

您可以在相同叢集中部署不同執行個體類型的 Amazon DocumentDB 複本。某個使用案例範例，即以較大的執行個體類型提供臨時分析流量，讓複本經久耐用。如果您部署一組混合的執行個體類型，請務必設定每個執行個體的容錯移轉優先順序。這有助於確保容錯移轉事件一律提升足夠大小的複本，以處理您的寫入負載。

- 復原

Amazon DocumentDB 會在寫入數據時持續備份您的數據。它可在 1-35 天的可設定期間內提供 point-in-time 復原 (PITR) 功能，稱為備份保留期。預設備份保留期為一天。Amazon DocumentDB 也會自動建立儲存磁碟區的每日快照，並在設定的備份保留期內保留這些快照。

如果您想要在備份保留期限之後保留快照，也可以隨時使用 AWS Management Console AND AWS Command Line Interface (AWS CLI) 啟動手動快照。如需詳細資訊，請參閱 [在 Amazon DocumentDB 中備份和還原](#)。

規劃遷移時請考慮以下事項：

- 選擇符合復原點目標 (RPO) 的 1-35 天備份保留期。
- 決定是否需要手動快照，如果需要，間隔為何。

遷移方法

將資料遷移到亞馬遜文件資料庫有三種主要方法。

Note

雖然您可以隨時在 Amazon DocumentDB 中建立索引，但是在匯入大型資料集之前，建立索引的整體速度會比較快。最佳實務是，我們建議您針對以下每種方法，先在 Amazon DocumentDB 中建立索引，然後再執行移轉。要做到這一點，您可以使用 [Amazon DocumentDB 索引工具](#)。

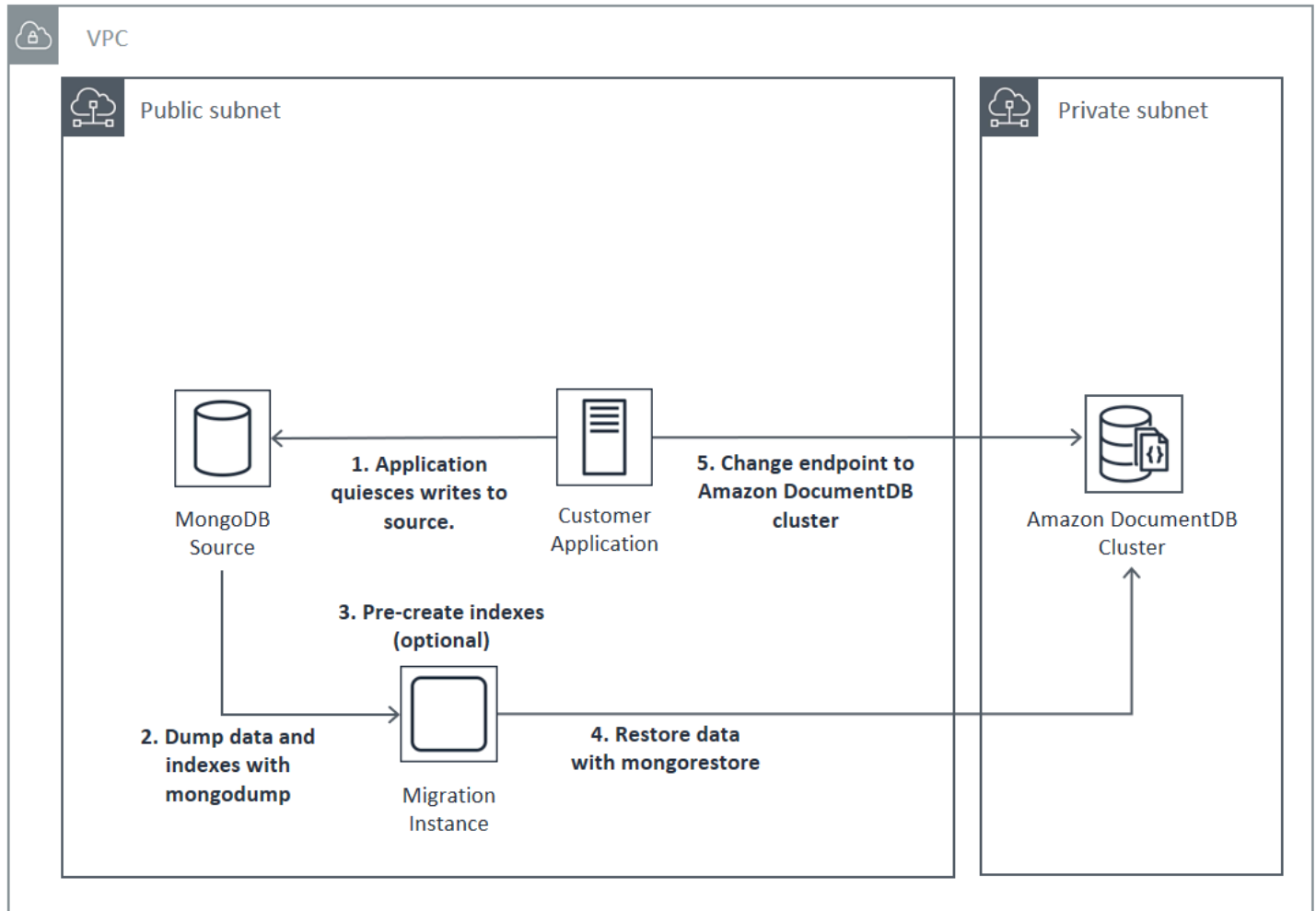
離線

離線方法使用mongodump和mongorestore工具將您的資料從來源 MongoDB 部署遷移到 Amazon DocumentDB 叢集。離線方法是最簡單的遷移方法，但也會產生最大的叢集停機時間。

離線遷移的基本程序如下：

1. Quiesce 寫入到您的 MongoDB 來源。
2. 傾印來自來源 MongoDB 部署的收集資料和索引。
3. 如果您要遷移到彈性叢集，請使用sh.shardCollection()命令建立您的分片集合。如果您要移轉至執行個體型叢集，請跳至下一個步驟。
4. 將索引還原到 Amazon DocumentDB 叢集。
5. 將收集資料還原至 Amazon DocumentDB 叢集。
6. 將您的應用程式端點變更為寫入 Amazon DocumentDB 叢集。

Offline Migration Approach



線上

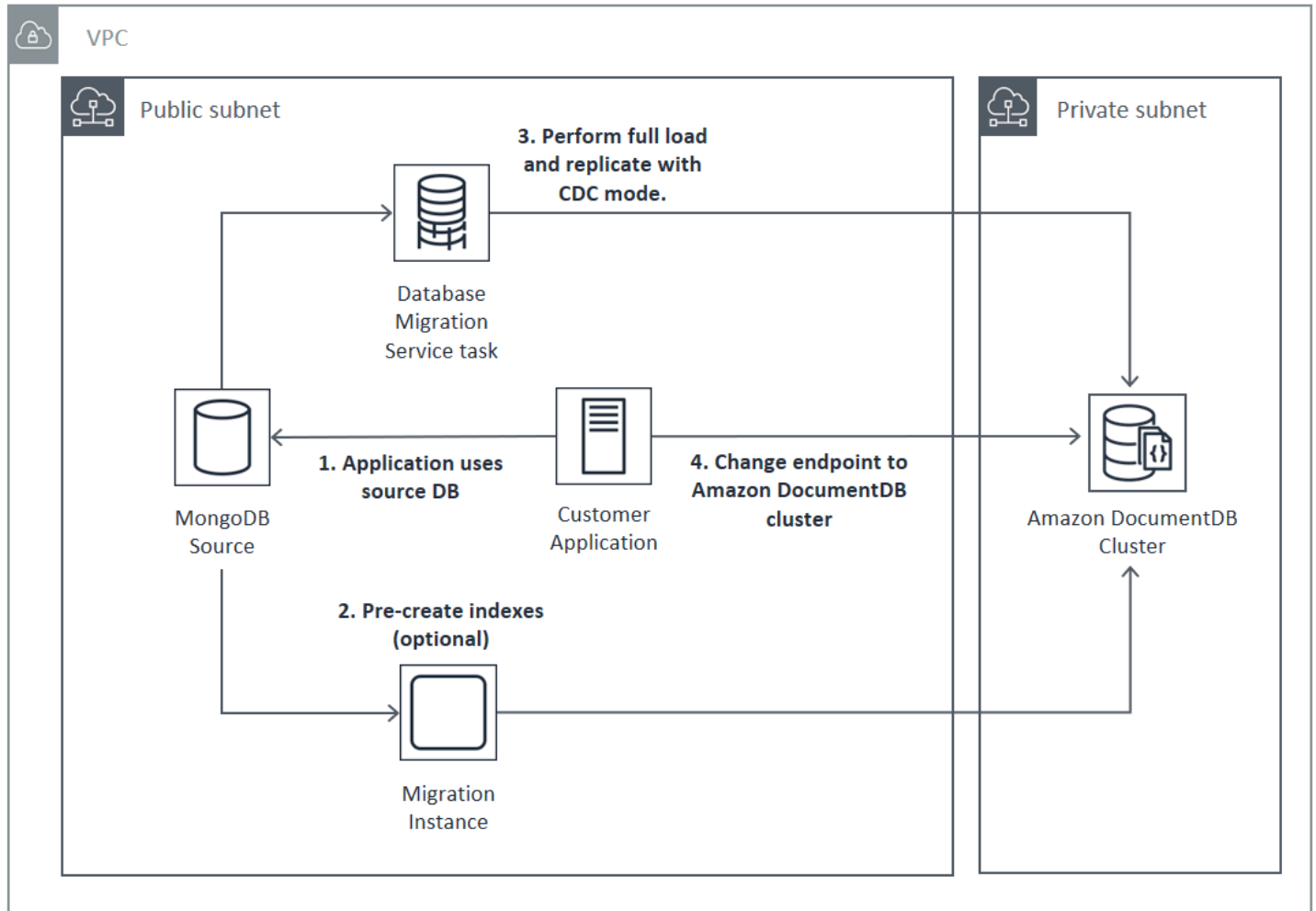
「線上」方法使用 AWS Database Migration Service (AWS DMS)。它會執行完整的資料負載，從您的來源 MongoDB 部署到您的 Amazon DocumentDB 叢集。然後，它會切換以將資料擷取 (CDC) 模式變更到複本變更。線上方法可減少叢集的停機時間，但它是三種方法中最慢的。

線上遷移的基本程序如下：

1. 您的應用程式通常會使用來源資料庫。
2. 如果您要遷移到彈性叢集，請使用 `sh.shardCollection()` 命令建立您的分片集合。如果您要移轉至執行個體型叢集，請跳至下一個步驟。
3. 在 Amazon DocumentDB 叢集中預先建立索引。
4. 建立 AWS DMS 任務以執行完全負載，然後啟用 CDC 從來源 MongoDB 部署到 Amazon DocumentDB 叢集。

5. 在 AWS DMS 任務完成完整負載並將變更複寫到 Amazon DocumentDB 之後，請將應用程式的端點切換到 Amazon DocumentDB 叢集。

Online Migration Approach



如需有關使用移 AWS DMS 轉的詳細資訊，請參閱[使用 Amazon DocumentDB 做為目標 AWS Database Migration Service](#)和[使用AWS Database Migration Service者指南](#)中的相關教學課程。

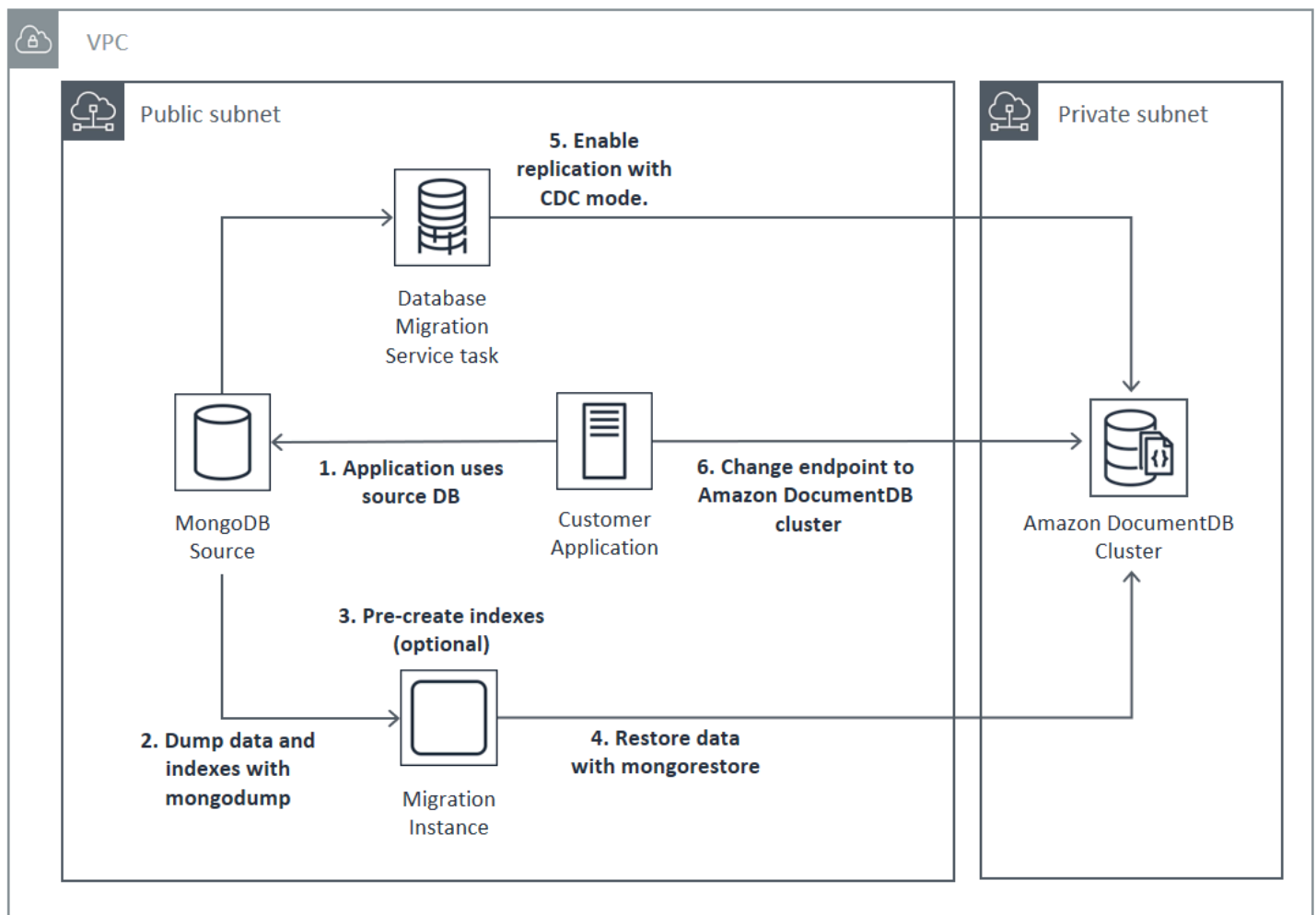
混合

混合式方法會使用mongodump和mongorestore工具，將您的資料從來源 MongoDB 部署遷移到 Amazon DocumentDB 叢集。然後，它會 AWS DMS 在 CDC 模式下使用複寫變更。混合方法可平衡遷移速度和停機時間，但它是三種方法中最複雜的一種。

混合遷移的基本程序如下：

1. 您的應用程式通常會使用來源 MongoDB 部署。
2. 傾印來自來源 MongoDB 部署的收集資料和索引。
3. 將索引還原到 Amazon DocumentDB 叢集。
4. 如果您要遷移到彈性叢集，請使用 `sh.shardCollection()` 命令建立分片集合。如果您要移轉至執行個體型叢集，請跳至下一個步驟。
5. 將收集資料還原至 Amazon DocumentDB 叢集。
6. 建立 AWS DMS 任務，以啟用 CDC 從來源 MongoDB 部署到 Amazon DocumentDB 叢集。
7. 當 AWS DMS 任務在可接受的視窗內複寫變更時，請變更應用程式端點以寫入 Amazon DocumentDB 叢集。

Hybrid Migration Approach



⚠ Important

AWS DMS 工作目前只能移轉單一資料庫。如果您的 MongoDB 來源有大量的資料庫，您可能需要自動化遷移任務的建立過程，或考慮使用離線方法。

無論您選擇哪種遷移方法，在遷移資料之前，在 Amazon DocumentDB 叢集中預先建立索引都是最有效率的。這是因為 Amazon DocumentDB 索引是以 parallel 方式插入資料，但是在現有資料上建立索引是單執行緒作業。

因為 AWS DMS 不會移轉索引 (只有您的資料)，因此不需要額外的步驟來避免再次建立索引。

遷移來源

如果您的 MongoDB 來源是獨立的 mongo 程序，而您想要使用線上或混合遷移方法，請先將您的獨立 mongo 轉換到複本集，以建立 oplog 做為 CDC 來源使用。

如果要從 MongoDB 複本集或碎片叢集遷移，請考慮為每個複本集或碎片建立鏈結或隱藏的次要複本，做為遷移來源使用。執行資料傾印可以強制記憶體產生工作集資料，影響生產執行個體的效能。您可以從不提供生產資料的節點遷移，以降低此風險。

遷移來源版本

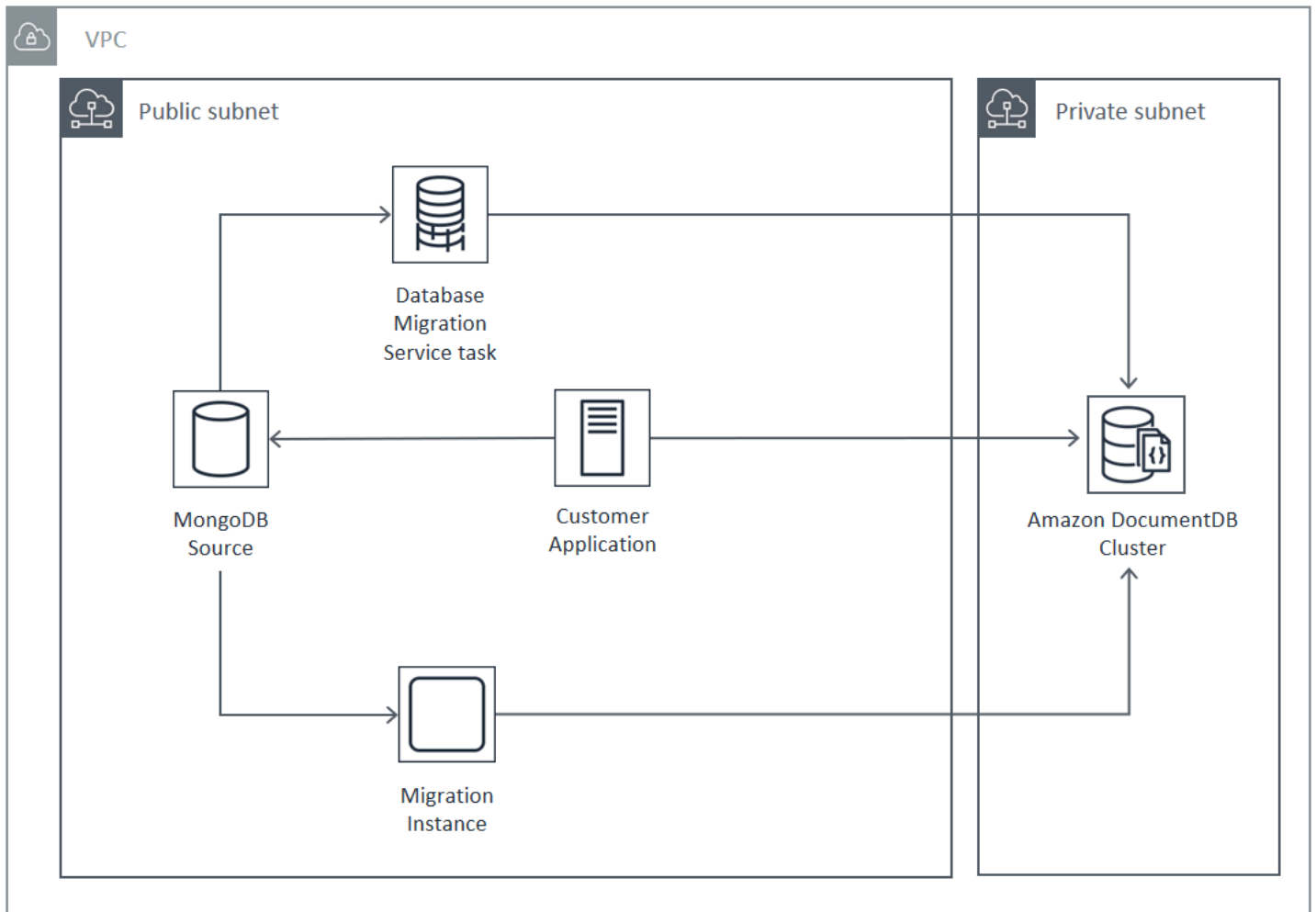
如果您的來源 MongoDB 資料庫版本與目的地 Amazon DocumentDB 叢集的相容性版本不同，您可能需要採取其他準備步驟以確保移轉成功。遇到的兩個最常見的需求是需要將來源 MongoDB 安裝升級到支援的移轉版本 (MongoDB 3.0 或更高版本)，以及升級您的應用程式驅動程式以支援目標 Amazon DocumentDB 版本。

如果您的遷移有這兩種需求的其中之一，遷移計畫中請務必包含這些步驟，以升級和測試任何的驅動程式變更。

遷移連線能力

您可以從在資料中心執行的來源 MongoDB 部署，或從 Amazon EC2 執行個體上執行的 MongoDB 部署遷移至亞馬遜文件資料庫。從在 EC2 執行的 MongoDB 遷移直截了當，只需要正確設定安全群組和子網路即可。

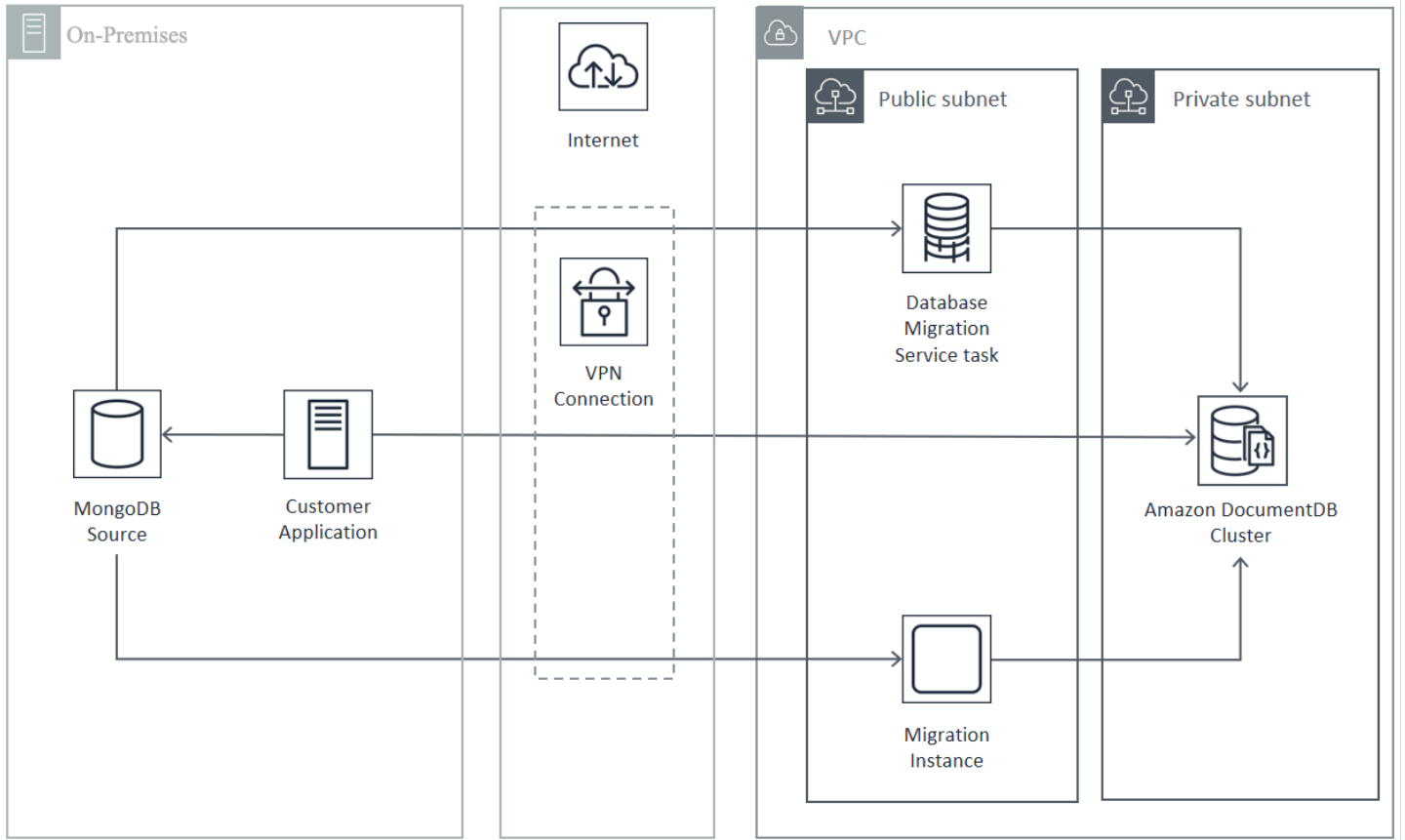
Migrating from EC2 Source



從現場部署資料庫遷移，您的 MongoDB 部署和虛擬私有雲端 (VPC) 之間需要有連線。您可以通過虛擬私人網絡 (VPN) 連接或使用該 AWS Direct Connect 服務來完成此操作。雖然您可以透過網際網路遷移到您的 VPC，但就安全觀點而言，這是最不恰當的連線方法。

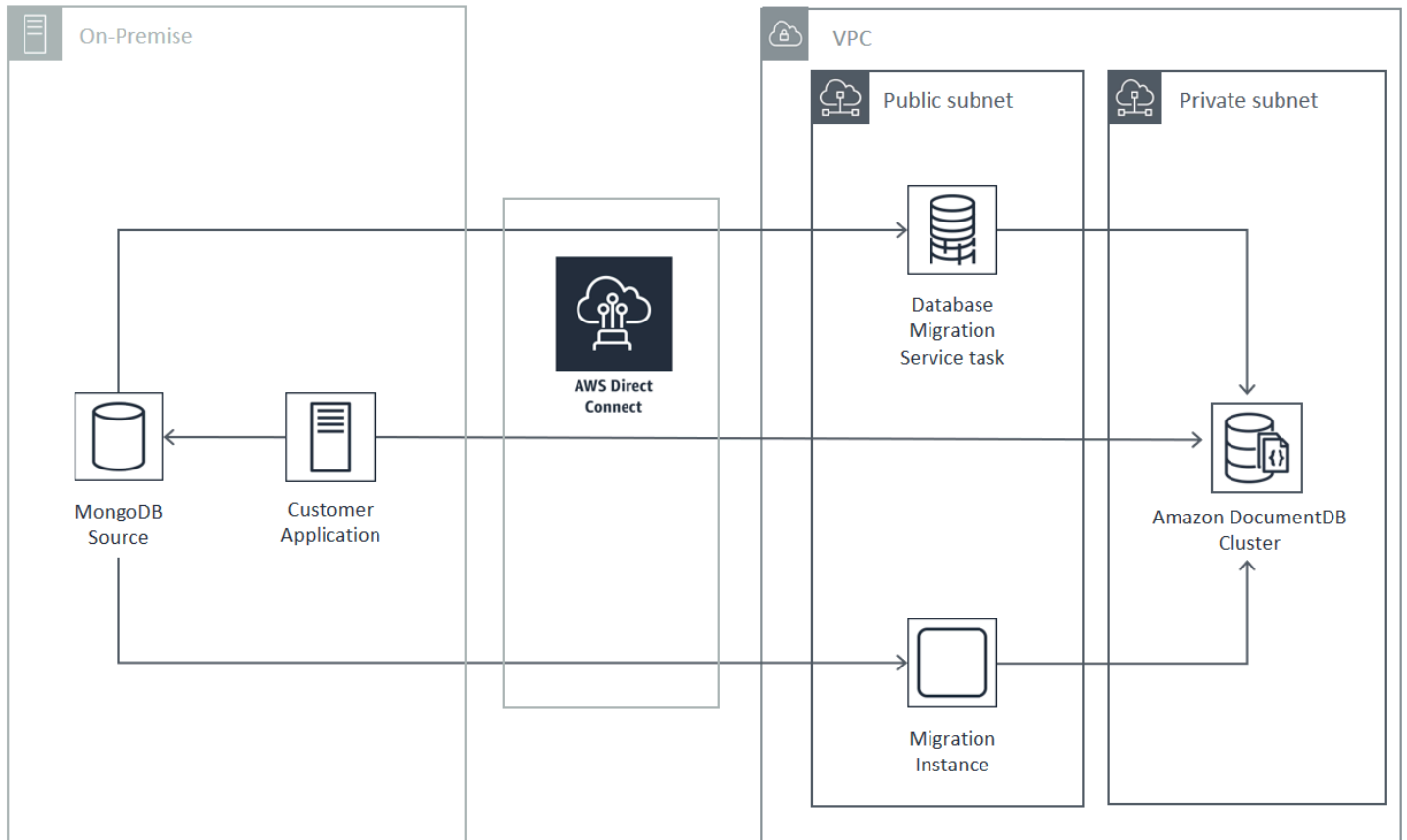
下圖說明透過 VPN 連線從現場部署來源遷移至 Amazon DocumentDB 料庫。

Migrating from On-Premise Source (VPN)



以下代表使用從現場部署來源遷移至 Amazon DocumentDB 料庫。AWS Direct Connect

Migrating from On-Premise Source (Direct Connect)



線上和混合移轉方法需要使用執行個體 AWS DMS 體，執行個體必須在 Amazon VPC 中的 Amazon EC2 上執行。所有方法都要求遷移伺服器執行 `mongodump` 和 `mongorestore`。通常，在啟動 Amazon DocumentDB 叢集的 VPC 中的 Amazon EC2 執行個體上執行遷移伺服器會比較容易，因為它大幅簡化了與 Amazon DocumentDB 叢集的連線能力。

測試

以下是預先遷移測試的目標：

- 驗證您所選擇的方法能否達成您想得到的遷移結果。
- 驗證您的執行個體類型和讀取偏好選項能否滿足您的應用程式效能需求。
- 驗證容錯移轉期間的應用程式行為。

遷移計畫測試注意事項

測試 Amazon DocumentDB 遷移計畫時，請考慮下列事項。

主題

- [還原索引](#)
- [傾印資料](#)
- [還原資料](#)
- [Oplog 大小](#)
- [AWS Database Migration Service 配置](#)
- [從碎片叢集遷移](#)

還原索引

根據預設，`mongorestore` 會建立傾印集合索引，但是在還原資料後才建立它們。在將資料還原到叢集之前，在 Amazon DocumentDB 中建立索引會比較快速。這是因為索引操作是在資料載入期間是平行作業。

如果您選擇預先建立索引，使用 `mongorestore` 還原資料時可提供 `--noIndexRestore` 選項，略過索引建立步驟。

傾印資料

`mongodump` 工具是從來源 MongoDB 部署傾印資料的慣用方法。視遷移執行個體的可用資源而定，您或許可以使用 `--numParallelCollections` 選項，將平行連線數量從預設值 4 往上增加，以加速 `mongodump`。

還原資料

此 `mongorestore` 具是將傾印的資料還原到 Amazon DocumentDB 執行個體的偏好方法。您可以在使用 `--numInsertionWorkersPerCollection` 選項還原期間，增加每個集合的工作者數目，提升還原效能。Amazon DocumentDB 叢集主要執行個體上的每個 vCPU 有一個工作者是開始使用的好地方。

Amazon DocumentDB 目前不支持該 `mongorestore` 工具的 `--oplogReplay` 選項。

根據預設，`mongorestore` 會略過插入錯誤，繼續還原程序。如果您要將不受支援的資料還原到 Amazon DocumentDB 執行個體，就會發生這種情況。例如，如果您的文件包含具有空字串的金鑰或值，就會發生這種情況。如果發生還原錯誤時，您希望讓 `mongorestore` 操作徹底失敗，請使用 `--stopOnError` 選項。

Oplog 大小

MongoDB 操作日誌 (oplog) 是封閉集合，包含資料庫所有的修改資料。您可以對複本集或碎片成員執行 `db.printReplicationInfo()` 操作，檢視 oplog 的大小和其包含的時間範圍。

如果您使用的是線上或混合方法，請確定每個複本集或碎片上的 oplog 都足夠大，可容納整個資料移轉程序期間所做的所有變更 (無論是透過 `mongodump` 或 AWS DMS 工作滿載)，以及合理的緩衝區。如需詳細資訊，請參閱 MongoDB 文件中的檢查 Oplog 大小。記錄 `mongodump` 或 `mongorestore` 程序或 AWS DMS 全部負載任務的第一個測試回合經歷時間，判斷所需的 oplog 大小下限。

AWS Database Migration Service 配置

[AWS Database Migration Service 使用者指南](#) 涵蓋將 MongoDB 來源資料遷移到 Amazon DocumentDB 叢集所需的元件和步驟。以下是用 AWS DMS 來執行線上或混合移轉的基本程序：

若要使用執行移轉 AWS DMS

1. 建立 MongoDB 來源端點。如需詳細資訊，請參閱 [使用 MongoDB 做為 AWS DMS 來源](#)。
2. 建立 Amazon DocumentDB 目標端點。如需詳細資訊，請參閱 [使用 AWS DMS 端點](#)。

如果您將目標端點設定為彈性叢集，請注意，您現有的 Amazon DocumentDB SSL 憑證無法與彈性叢集搭配使用，因此您需要使用下列步驟將新的 SSL 憑證附加到端點：

- a. 訪問 <https://www.amazontrust.com/repository/SFSRootCAG2.pem> 並將內容保存為「根據」文件。這是您在後續步驟中需要匯入的憑證檔案。
 - b. 建立彈性叢集端點時，在端點組態下，選擇新增 CA 憑證。
 - 然後，對於憑證識別碼，輸入 `SFSRootCAG2.pem`。
 - 對於 Import certificate file (匯入憑證檔案)，請選擇 Choose file (選擇檔案)，然後瀏覽至您先前下載的 `SFSRootCAG2.pem` 檔案。選取並開啟檔案。選擇匯入憑證，然後 `SFSRootCAG2.pem` 從選擇憑證下拉式清單中選擇。
3. 建立至少一個 AWS DMS 複製執行個體。如需詳細資訊，請參閱 [使用 AWS DMS 複製執行個體](#)。
 4. 至少建立一個 AWS DMS 複製工作。如需詳細資訊，請參閱 [使用 AWS DMS 任務](#)。

針對線上遷移，您的遷移任務會使用遷移類型 Migrate existing data and replicate ongoing changes (遷移現有資料及複寫持續的變更)。

針對混合遷移，您的遷移任務會使用遷移類型 Replicate data changes only (僅複寫變更的資料)。您可以選擇 CDC 開始時間，以符合您 mongodump 操作的傾印時間。MongoDB oplog 是等冪的。為避免遺失變更，mongodump 結束時間和 CDC 開始時間之間最好能夠重疊幾分鐘。

從碎片叢集遷移

將資料從 MongoDB 資料分割叢集遷移到 Amazon DocumentDB 執行個體的程序，基本上是多個 parallel 移轉複本集的程序。測試碎片叢集遷移時的重要考量是，有些碎片的使用量可能比其他碎片更多。這種情況會導致資料遷移的經歷時間不同。確保在規劃和測試時評估每個碎片的 oplog 需求。

以下是遷移碎片叢集時，需要考慮的一些組態問題：

- 執行 mongodump 或開始 AWS DMS 遷移任務之前，您必須停用碎片叢集平衡器，等待任何正在進行的遷移完成。如需詳細資訊，請參閱 MongoDB 文件的停用平衡器。
- 如果您使用複製 AWS DMS 資料，請在執行移轉工作之前，在每個碎片上執行 cleanupOrphaned 指令。如果您不執行此命令，任務可能會因為重複的文件 ID 而失敗。請注意，此命令可能會影響效能。如需詳細資訊，請參閱 MongoDB 文件中的 cleanupOrphaned。
- 如果您使用 mongodump 工具傾印資料，應該每個碎片執行一次 mongodump 程序。最有時間效率的方法可能需要多部遷移伺服器，將傾印效能提升到最高。
- 如果您使 AWS Database Migration Service 用複製資料，則必須為每個碎片建立來源端點。同時針對您要遷移的每個碎片，至少執行一項遷移任務。最有時間效率的方法可能需要多個複寫執行個體，將遷移效能提升到最高。

效能測試

成功將資料移轉到測試 Amazon DocumentDB 叢集後，請對叢集執行測試工作負載。透過 Amazon CloudWatch 指標驗證您的效能是否符合或超過 MongoDB 來源部署目前的輸送量。

驗證下列關鍵的 Amazon DocumentDB 指標：

- 網路輸送量
- 寫入輸送量
- 讀取輸送量
- 複本延遲

如需詳細資訊，請參閱 [Amazon DocumentDB](#)。

容錯移轉測試

確認 Amazon DocumentDB 容錯移轉事件期間應用程式的行為是否符合您的可用性需求。若要在主控台上啟動 Amazon DocumentDB 叢集的手動容錯移轉，請在「叢集」頁面上，選擇「動作」功能表上的「容錯移轉」動作。

您也可以從 AWS CLI 執行 `failover-db-cluster` 操作，啟動容錯移轉。如需詳細資訊，請參閱參考 [failover-db-cluster](#) 資料的 Amazon DocumentDB 一節。AWS CLI

其他資源

請參閱《AWS Database Migration Service 使用指南》中的下列主題：

- [使用 Amazon DocumentDB 作為目標 AWS Database Migration Service](#)
- [逐步解說：從 MongoDB 遷移到 Amazon DocumentDB](#)

遷移教戰手冊：MongoDB 到 Amazon DocumentDB

此移轉教戰手冊提供資源和步驟，協助您從 MongoDB 資料庫遷移到 Amazon DocumentDB。

遷移程序

以下列出的是將資料從 MongoDB 資料庫遷移到 Amazon DocumentDB 時通常涉及的高階步驟。

主題

- [步驟 1：兼容性和功能差異](#)
- [步驟 2：概念證明](#)
- [步驟 3：移轉資料](#)
- [步驟 4：資料驗證](#)
- [步驟 5：應用程式切換](#)

步驟 1：兼容性和功能差異

Amazon DocumentDB 與阿帕奇 2.0 的開源 MongoDB 3.6，4.0 和 5.0 的 API 進行交互。因此，您可以在 Amazon DocumentDB 中使用相同的 MongoDB 驅動程式、應用程式和工具，只需進行少量變更或完全沒有變更。

第一個步驟是檢查應用程式在 MongoDB 資料庫上使用的運算子和索引之間的相容性，以及它們在 Amazon DocumentDB 中的可用性，以及瞭解它們之間的功能差異。

運營商兼容

使用 [Amazon DocumentDB 相容性工具](#) *，輕鬆探索應用程式是否在其查詢中使用任何不受支援的運算子。此工具可以掃描 MongoDB 資料庫伺服器記錄檔或應用程式原始程式碼，以提供不受支援的運算子的報告。如果您發現不受支援的運算子的使用，您需要修改應用程式以解決不受支援的運算子。

若要檢查設定中使用的 MongoDB 運算子與支援的 Amazon DocumentDB 運算子之間的相容性，請執行下列指令：

```
git clone https://github.com/aws-labs/amazon-documentdb-tools.git
cd amazon-documentdb-tools/compat-tool/
python3 compat.py --version <Amazon DocumentDB version> --directory <mongodb logfile/
source code>
```

如需詳細資訊，請參閱 [支援的 MongoDB API、營運和資料類型](#)。

* 未正式支援 AWS。

索引相容性

您可以使用 [Amazon DocumentDB 索引工具](#) * 來確定您是否正在使用 Amazon DocumentDB 中不支持的任何索引類型。此工具需要與來源資料庫的連線，才能讀取索引定義。

為此，您首先需要使用 `--dump-indexes` 選項將索引定義轉儲到目錄中。然後使用該 `--show-issues` 選項運行該工具，提供目錄以查找不兼容的索引。

匯出索引：

```
git clone https://github.com/aws-labs/amazon-documentdb-tools.git
sudo pip install -r amazon-documentdb-tools/index-tool/requirements.txt
mkdir <directory to dump index definitions>
```

```
python3 migrationtools/documentdb_index_tool.py --dump-indexes --dir <directory> --uri  
<source-mongodb-uri>
```

檢查不相容的索引：

```
python3 migrationtools/documentdb_index_tool.py --show-issues --dir <dumped-index-  
definitions-directory>
```

如果您發現任何不受支援的索引類型的用法，您必須修改應用程式或資料模型，以解決問題或繼續，而不使用不相容的索引。

如需 Amazon DocumentDB 中支援的索引類型和屬性的詳細資訊，請參閱[索引和索引屬性](#)和[如何在 Amazon DocumentDB 上建立索引](#)。

* 未正式支援AWS。

功能差異

檢[MongoDB 的功能差異](#)閱以熟悉差異。

步驟 2：概念證明

透過在 Amazon DocumentDB 上執行應用程式或常規測試套件來測試功能和效能，以執行概念證明。您可能需要在 Amazon DocumentDB 叢集中填入資料，才能執行測試。例如，您可以使用mongodump和mongorestore工具從來源 MongoDB 複製資料。

功能測試

建立 Amazon DocumentDB 叢集 (請參閱[創建一個 Amazon DocumentDB 集群](#)) 並執行您的應用程式或功能測試套件，以驗證所有應用程式工作流程是否在 Amazon DocumentDB 上繼續無縫運作。

性能測試

在 Amazon DocumentDB 上執行的應用程式或效能測試套件上執行效能測試，其工作負載類似於您的生產工作負載，以查看設定是否符合延遲需求。微調您的工作負載以提高效能，或視情況擴展 Amazon DocumentDB 叢集。如需詳細資訊，請參閱[效能與資源使用率](#)及[擴展 Amazon DocumentDB 集群](#)。

請務必使用正確的執行個體類型來調整 Amazon DocumentDB 叢集的大小，以獲得最佳效能。如需詳細資訊，請參閱[執行個體大小設定](#)。

您可以使用[Amazon 文件資料庫大小計算器](#) * 來協助您估算 Amazon DocumentDB 叢集的大小。

* 未正式支援AWS。

容錯移轉測

您可能想要觀察應用程式如何回應 Amazon DocumentDB 主節點重新啟動、主節點容錯移轉或多節點叢集中的主節點刪除，以及重新開機或移除複本節點的時間。這可協助您確認應用程式對這些事件有彈性。如需詳細資訊，請參閱 [測試容錯移轉](#)。

若要瞭解應用程式應該容忍的例外狀況以及如何有效處理這些例外情況，請參閱使用 [Amazon DocumentDB 建置彈性應用程式](#)。

Note

在 Amazon DocumentDB 上測試工作負載沒有替代品

步驟 3：移轉資料

成功完成概念證明之後，請將您的資料遷移到 Amazon DocumentDB。我們的大多數客戶都使用線上或離線遷移方法來遷移他們的資料。

線上移轉

使用線上遷移方法，您可以將資料從幾 GB 到數 TB 的來源資料庫遷移到幾乎零停機時間的 Amazon DocumentDB。如需詳細資訊，請參閱 [AWS Database Migration Service \(AWS DMS\)](#)。

如果您從 MongoDB 數據庫遷移，則可以使用AWS DMS完全負載並複製正在進行的更改。

如需 step-by-step 程序，請參閱[使用線上方法移轉至 Amazon DocumentDB](#)。

您可以在使用AWS Database Migration Service者指南中的 [〈使用 Amazon DocumentDB 作為目標〉—AWS Database Migration Service](#)節中找到其他資訊。

注意事項AWS DMS：

- 分段：使用移轉多 TB 資料庫時AWS DMS，預設設定可能會很慢，因為預設情況下，DMS 的完整負載為每個集合的單執行緒，因此移轉時間更長。若要加快大型資料庫移轉的完整負載，您可以使用中的分段功能AWS DMS。

如需如何搭配使用區段的詳細資訊AWS DMS，請參閱[搭配使用自動分段AWS DMS](#)。

- DMS 執行個體類型：若要加速資料移轉，您必須[選擇正確的 DMS 執行個體](#)。

離線遷移

離線遷移是將資料庫移至 Amazon DocumentDB 最直接的方法。此方法主要用於 POC 以及在移轉期間可能需要寫入停機時間的工作負載。

如需 step-by-step 程序，請參閱使用離線方法[從 MongoDB 遷移至 Amazon DocumentDB 料庫](#)。

步驟 4：資料驗證

成功遷移資料後，請驗證資料的正確性以獲得信心。在 AWS DMS 移轉工作主控台上，您可以找到移轉的資料指標。如需詳細資訊，請參閱[驗證移轉的資料](#)。

您也可以使用 [Amazon DocumentDB DataDiffer 工具](#) * 來驗證來源集合和目標集合之間的資料一致性。

* 未正式支援 AWS。

步驟 5：應用程式切換

這包括將應用程式的資料庫連接字串變更為使用 Amazon DocumentDB 叢集。

如需連線至 Amazon 文件資料庫的詳細資訊，請參閱 [以複本集形式連接至 Amazon DocumentDB](#)

線上移轉

完成完整的資料載入之後，會 AWS DMS 繼續將持續的變更從您的來源複寫到 Amazon DocumentDB。在變更被趕上並完成資料驗證檢查之後，您可以對 Amazon DocumentDB 執行切換。

離線移轉

完成完整的資料載入和資料驗證檢查之後，您就可以執行 Amazon DocumentDB 的切換作業。

其他資源

以下是一些可以幫助您遷移的其他資源：

- 影片：[遷移至 Amazon DocumentDB 的最佳實務](#)
- 影片：[開始使用 Amazon DocumentDB 可觀察性和監控](#)
- 其他實用程序：[Amazon DocumentDB 工具](#) *
- 移轉開發人員指南：[遷移到 Amazon DocumentDB](#)

* 未正式支援AWS。

Amazon DocumentDB 就地主要版本升級

Amazon DocumentDB 只有在經過廣泛測試之後，才能使用新版本的資料庫引擎。您可以選擇將 Amazon DocumentDB 叢集升級到新版本的方式和時機。

目前，Amazon DocumentDB 支持三個主要版本：Amazon DocumentDB 3.6，4.0 和 5.0。您可以執行資料庫的就地主要版本升級 (MVU)，同時保留叢集的相同端點、儲存區和標籤，並且可以繼續使用您的應用程式而無需進行任何修改。此功能在提供 Amazon DocumentDB 5.0 的所有區域均可免費使用。

Important

在就地主要版本升級期間，您的 Amazon DocumentDB 叢集將無法使用，而且您的叢集將會遇到多次重新啟動。根據集合、索引、資料庫和執行個體的數目，升級停機時間會因叢集而異。我們建議您在維護時段或使用率低時段執行升級。叢集升級後，您無法將叢集降級為舊版，但您可以選擇將升級前快照還原為新叢集。

主題

- [先決條件和限制](#)
- [就地主要版本升級的最佳作法](#)
- [執行就地主要版本升級](#)
- [疑難排解就地主要版本升級](#)
- [Amazon DocumentDB 3.6/4.0 到 5.0 升級的集群和新的 Amazon DocumentDB 5.0 集群之間的差異](#)

先決條件和限制

以下是您在執行升級之前可能需要瞭解並採取行動的就地主要版本升級的先決條件和限制：

- 執行個體類型 — Amazon DocumentDB 4.0/5.0 不支援 r4.* 執行個體。若要繼續就地主要版本升級，請將 r4.* 執行個體修改為 r5.* 執行個體。如需詳細資訊，請參閱[修改 Amazon DocumentDB 實例](#)。如需以 Amazon DocumentDB 引擎版本[各區域的支援執行個體類別](#)為基礎的支援執行個體，請參閱。
- 執行個體作業系統修補程式 — 就地主要版本升級需要最新的作業系統 (OS) 修補程式才能繼續。在繼續就地升級之前，請對執行個體套用任何擱置中的作業系統維護動作。如需詳細資訊，請參閱[使用強制作業系統更新](#)。

Note

在某些情況下，如果您有擱置中的叢集層級引擎修補程式，則看不到執行個體 OS 修補程式。您可能需要先套用叢集層級引擎修補程式，才能繼續套用執行個體作業系統修補程式，然後再進行就地主要版本升級。請參閱[對叢集的引擎版本執行修補程式更新](#)。

- 在提供 Amazon DocumentDB 5.0 的所有區域均提供就地主要版本升級。
- 使用 Amazon DocumentDB 4.0 做為目標版本時，不支援就地主要版本升級。
- Amazon DocumentDB 全域叢集和彈性叢集目前不支援就地主要版本升級。

Note

若要升級全域叢集，請從全域叢集中刪除次要叢集、將主要叢集轉換為地區叢集、在地區(主要)叢集上執行就地主要版本升級，然後使用相同名稱新增次要叢集來重新建立全域叢集，以保留與先前相同的端點。請注意，當您升級的主要叢集將資料複寫到新增的次要叢集時，會產生 IO 費用。如需如何在刪除之前從全域叢集移除次要叢集的詳細步驟，請參閱[從 Amazon DocumentDB 全域叢集中移除叢集](#)。

- 如果您有大量的索引 (>10,000) 並且在較小的執行個體上操作 (例如，t3.medium)，則必須將主要執行個體擴展至較大的執行個體 (例如，至少 r5.xlarge)，才能在執行個體中保留足夠的記憶體來執行就地主要版本升級。您可以選擇在就地主要版本升級完成後縮小例證大小。請參閱下表，瞭解每個執行個體類型就地主要版本升級所支援的索引數目上限：

對於記憶體最佳化執行個體 (db.r5.*)：

| 執行個體 | 就地 MVU 支援的最大索引 |
|---------------|----------------|
| db.r5.large | 10 萬 |
| db.r5.xlarge | 2 萬 |
| db.r5.2xlarge | 30 萬 |
| db.r5.4xlarge | 400 萬 |
| db.r5.8xlarge | 5 萬 |

| 執行個體 | 就地 MVU 支援的最大索引 |
|----------------|----------------|
| db.r5.12xlarge | 700 萬 |
| db.r5.16xlarge | 800 萬 |
| db.r5.24xlarge | 1 米 |

對於高載性能實例 (db.t3 , db.t4g)

| 執行個體 | 就地 MVU 支援的最大索引 |
|---------------|----------------|
| db.t4g.medium | 3 公里 |
| db.t3.medium | 10K |

對於記憶體最佳化的重力子例證 (db.r6g.*) :

| 執行個體 | 就地 MVU 支援的最大索引 |
|-----------------|----------------|
| db.r6g.large | 10 萬 |
| db.r6g.xlarge | 2 萬 |
| db.r6g.2xlarge | 30 萬 |
| db.r6g.4xlarge | 400 萬 |
| db.r6g.8xlarge | 5 萬 |
| db.r6g.12xlarge | 700 萬 |
| db.r6g.16xlarge | 800 萬 |

Note

如果您有超過 1M 的索引，請聯絡 AWS 支援人員，不要繼續進行就地主要版本升級。

就地主要版本升級的最佳作法

使用複製的叢集就地測試主要版本升級

1. 若要測試就地主要版本升級，建議您使用快速複製功能來建立目標叢集的複製。除非您修改叢集上的任何資料，否則在複製的磁碟區上測試就地主要版本升級時，不會產生任何儲存體費用。如需磁碟區複製的詳細資訊，請參閱[複製 Amazon DocumentDB 叢集的磁碟區](#)。
2. 若要更實際地估計完成就地主要版本升級所花費的時間，請將複製叢集的執行個體計數與目標叢集相符。
3. 我們建議您完全測試新升級的 Amazon DocumentDB 5.0 叢集是否有任何功能差異，以確保一切正常運作。

在就地主要版本升級之前

1. 準備好與版本相容的叢集參數群組。

針對新引擎版本使用 Amazon DocumentDB 預設叢集參數群組，或為新引擎版本建立自己的自訂叢集參數群組。

如果您將 Amazon DocumentDB 叢集參數群組關聯為升級請求的一部分，就地主要版本升級將自動重新開機叢集以套用新的參數群組。

2. 請確定您已符合 [必要條件和限制] 區段中所述的就地主要版本升級的先決條件。
3. 建立手動快照。

升級程序會在升級期間建立資料庫叢集的快照集。強烈建議您在升級程序之前建立自己的手動快照。請參閱[建立手動叢集快照](#)。

Note

在就地主要版本升級完成後，升級程序建立的自動快照將不會自動刪除。只要在保留期內，此快照就不會產生任何費用。在確認成功升級叢集之後，您可以選擇刪除此快照。

快照命名為 `preupgrade-<name>-<version>-<timestamp>`。

| Snapshot identifier | Cluster identifier | Snapshot creation time | Status | Progress | VPC | Type |
|--|--------------------|----------------------------|-----------|-----------|----------------|-----------|
| preupgrade-example-cluster-3-6-0-to-5-0-0-2023-08-31-17-41 | example-cluster | 8/31/2023, 12:45:58 PM ... | available | Completed | vpc-02c0445... | manual |
| rds:preupgrade-example-cluster-3-6-0-to-5-0-0-2023-08-31-17-41 | example-cluster | 8/31/2023, 12:45:58 PM ... | available | Completed | vpc-02c0445... | automated |

4. 檢查您是否已排程叢集的就地主要版本升級。

如果您已修改叢集並選取在下一個維護時段中套用叢集，就地主要版本升級排程將不會顯示在主控台上，但您可以在 CLI 中檢視它。您可以執行下列命令來檢查是否已排程就地主要版本升級：

```
aws docdb describe-db-cluster \
--region $REGION \
--db-cluster-identifier $CLUSTER_NAME

"PendingModifiedValues": {
  "EngineVersion": "5.0.0"
},
```

- 在較低環境中使用磁碟區複製來執行多次乾式執行，以便在任何執行計畫和功能差異上就地主要版本升級後測試叢集。我們建議使用相同數量和大小的執行個體進行複製，以便更好地估計就地主要版本升級執行時間。如需詳細資訊，請參閱 [複製 Amazon DocumentDB 叢集的磁碟區](#)。
- 如果上一個步驟成功，請繼續在生產叢集上就地進行主要版本升級。

在就地主要版本升級期間

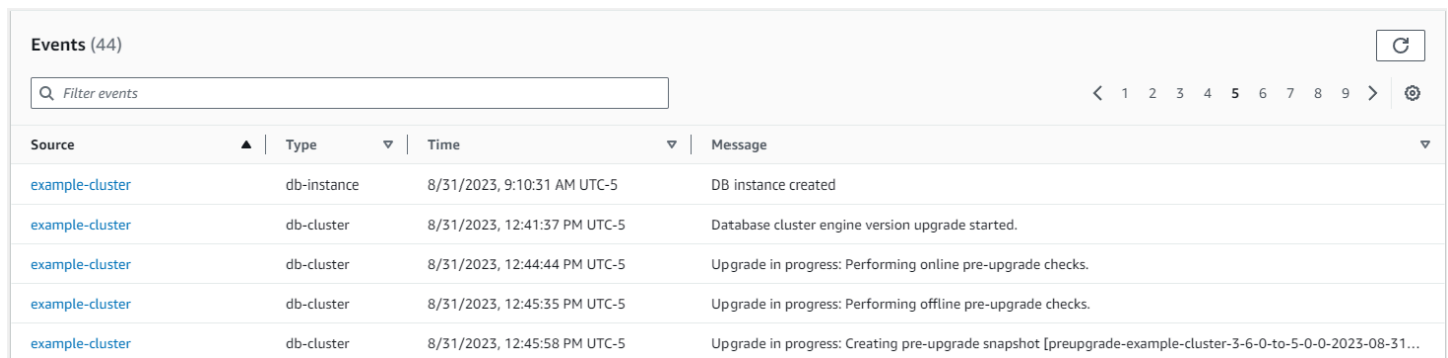
您可以透過訂閱叢集維護事件來監視就地主要版本升級的進度。升級完成後，您會收到「資料庫叢集主要版本已升級」事件。升級期間發生的這個事件和其他事件會出現在 Amazon DocumentDB 主控台中叢集詳細資料頁面的「事件和標籤」區段中。然後，叢集狀態會從「升級」變更為「可用」。

您可以從 CLI 執行 `aws docdb create-event-subscription` 以建立事件並監視 `aws docdb describe-events` 進度。您也可以針對上述事件設定至 Amazon SNS 的事件通知，做為透過電子郵件、推送訊息和其他方法通知的目標。如需詳細資訊，請參閱 [訂閱 Amazon 文件資料庫事件訂閱](#)。

就地主要版本升級會在升級期間產生下列事件：

- `<cluster-name><timestamp>` 升級進行中：建立升級前快照 [升級前--]
- 升級進行中：複製磁碟區。
- 升級進行中：升級寫入器。
- 升級進行中：升級讀者。
- 資料庫叢集主要版本已升級。

事件也會顯示在主控台的「事件」(Events) 頁面下方：



| Source | Type | Time | Message |
|-----------------|-------------|------------------------------|---|
| example-cluster | db-instance | 8/31/2023, 9:10:31 AM UTC-5 | DB instance created |
| example-cluster | db-cluster | 8/31/2023, 12:41:37 PM UTC-5 | Database cluster engine version upgrade started. |
| example-cluster | db-cluster | 8/31/2023, 12:44:44 PM UTC-5 | Upgrade in progress: Performing online pre-upgrade checks. |
| example-cluster | db-cluster | 8/31/2023, 12:45:35 PM UTC-5 | Upgrade in progress: Performing offline pre-upgrade checks. |
| example-cluster | db-cluster | 8/31/2023, 12:45:58 PM UTC-5 | Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-example-cluster-3-6-0-to-5-0-0-2023-08-31... |

在中 AWS CLI，您可以使用下列指令來追蹤進度：

```
aws docdb describe-events --source-identifier $CLUSTER_NAME --source-type db-cluster
{
  "Events": [
    {
      "SourceIdentifier": "mycluster",
      "SourceType": "db-cluster",
      "Message": "Database cluster engine version upgrade started.",
      "EventCategories": [
        "maintenance"
      ],
      "Date": "2023-07-11T23:20:32.444000+00:00",
      "SourceArn": "arn:aws:rds:us-east-1:xxxx:cluster:mycluster"
    }
  ]
}
```

在就地主要版本升級之後

對於 Amazon DocumentDB 3.6，請在叢集中新增一個標籤，以區分叢集已從 Amazon DocumentDB 3.6 升級到 Amazon DocumentDB 5.0，而不是新建立的 Amazon DocumentDB 5.0 叢集。請參閱有關已升級的 Amazon DocumentDB 5.0 叢集和新的 Amazon DocumentDB 5.0 叢集之間的差異章節。

如果您需要還原至升級後狀態，請在就地主要版本升級完成後建立手動快照。一旦就地主要版本升級完成，自動快照集程序將會繼續執行。只要手動快照在保留期間內，就不會產生任何費用。

若要使用與 Amazon DocumentDB 5.0 相關聯的新功能 (例如用戶端欄位層級加密)，我們建議您將驅動程式版本升級至 MongoDB 5.0 API 版本。如需詳細資訊，請參閱 [Amazon DocumentDB 5.0 中的新功能](#) 閱以取得 Amazon DocumentDB 5.0 功能的清單。

Important

在就地主要版本升級 (MVU) 執行之後，Amazon DocumentDB 5.0 叢集會立即重新填入索引中繼資料，並根據資料庫引擎優化查詢執行計劃的方式來重新填入索引中繼資料。在索引中繼資料重新計算程序完成之後，Amazon DocumentDB 叢集上的預期查詢效能將恢復。此程序通常會在幾分鐘內完成，但最多可持續兩小時，視叢集上的索引數目而定。

此外，在就地 MVU 之後，立即重新開機、容錯移轉或擴展/縮減寫入器執行個體，可能會中斷叢集上的索引中繼資料計算程序。就地 MVU 完成之後，我們建議您在 Amazon DocumentDB 5.0 叢集上觀察到預期的查詢效能後進行此類變更。

如果您在就地 MVU 之後看到此暫時性能下降 AWS 持續超過兩個小時，請聯繫支持人員。

完整測試升級後的 Amazon DocumentDB 5.0 叢集，以確保一切正常運作。

Note

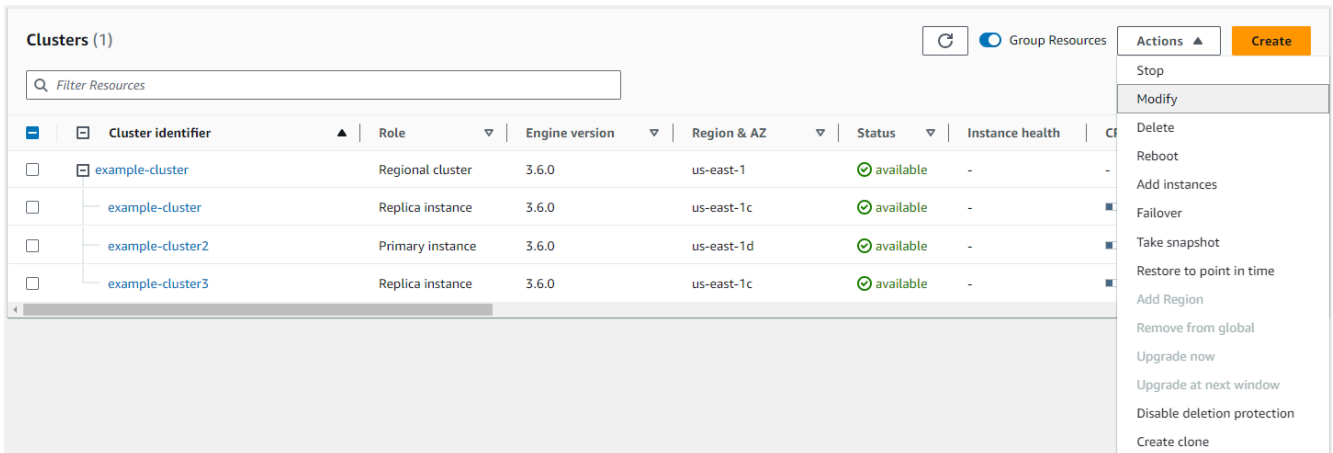
在啟用變更串流的 Amazon DocumentDB 叢集上執行就地 MVU 之後，會保留先前的變更串流事件，並且可以使用或繼續。resumeToken startAtOperationTime 與任何新建立的 Amazon DocumentDB 叢集一樣，如果記錄大小大於 51,200MB，則變更早於 change_stream_log_retention_duration 刪除的串流事件日誌。

執行就地主要版本升級

Using the AWS Management Console

若要使用下列項目執行就地主要版本升級 AWS Management Console：

1. 登入 [AWS Management Console](#) 並開啟 Amazon DocumentDB 主控台。
2. 在「叢集」表格中，選取來源叢集，按一下動作，然後按一下修改。



3. 在 [叢集規格] 區段的 [修改叢集] 對話方塊中，從 [引擎版本] 下拉式功能表中選擇目標資料庫版本 (5.0)。

Cluster specifications

Cluster identifier [Info](#)
Specify a unique cluster identifier.

Engine version
5.0.0 ▼
VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.
Select VPC security groups ▼
default (VPC) ✕
New master password [Info](#)
Confirm password [Info](#)
Password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).

- 在 [叢集選項] 區段中，選擇適當的叢集參數群組 (default.docdb5.0) 或自訂建立的參數群組。

Cluster options

Port
TCP/IP port that is used to connect to the cluster.
27017

Cluster parameter group
default.docdb5.0 ▼

[?](#) To create a new custom parameter group, please go to the Parameter group page, create your new custom parameter group and re-initiate the in-place Major Version Upgrade process.

- 完成後，向下捲動並選擇「繼續」。
- 在「排定修改」區段中，選擇您偏好的排程計劃：立即套用或在下一個維護時段套用。

然後選擇 Modify cluster (修改叢集)。

Modify cluster: example-cluster

Summary of modifications
You are about to submit the following modifications. Only values that will change are displayed. Carefully verify your changes and click Modify cluster.

| Attribute | Current value | New value |
|-------------------------|------------------|------------------|
| Cluster parameter group | default.docdb3.6 | default.docdb5.0 |
| Engine version | 3.6.0 | 5.0.0 |

Scheduling of modifications

When to apply modifications

Apply during the next scheduled maintenance window
Current maintenance window: fri:09:03-fri:09:33

Apply immediately
The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

Modifications will not be applied immediately
Modifications will be applied during the next scheduled maintenance window (fri:09:03-fri:09:33). To apply these modifications immediately, choose "Apply immediately" above.

Cancel Back **Modify cluster**

7. 在叢集表格中，記下叢集升級時的狀態：

Clusters (1) Group Resources Actions Create

Filter Resources

| Cluster identifier | Role | Engine version | Region & AZ | Status | Instance health | CPU | Current activity |
|--------------------|------------------|----------------|-------------|----------------|-----------------|--------|------------------|
| example-cluster | Regional cluster | 3.6.0 | us-east-1 | ⌚ upgrading... | - | - | - |
| example-cluster | Replica instance | 3.6.0 | us-east-1c | ⌚ upgrading... | - | 14.96% | 0 Connections |
| example-cluster2 | Primary instance | 3.6.0 | us-east-1d | ⌚ upgrading... | - | 13.54% | 0 Connections |
| example-cluster3 | Replica instance | 3.6.0 | us-east-1c | ⌚ upgrading... | - | 14.45% | 0 Connections |

Using the AWS CLI

使用具有所需引擎版本和allow-major-version-upgrade標誌集的 modify-db-cluster API：

```
aws docdb modify-db-cluster \
  --db-cluster-identifier $CLUSTER_NAME \
  --allow-major-version-upgrade \
  --engine-version 5.0 \
  --apply-immediately \
  --cluster-parameter-group $PARAMETER_GROUP \
  --region $REGION
```

疑難排解就地主要版本升級

如果發生故障，就地主要版本升級將嘗試復原升級，以假設開始升級之前叢集的最後一個作業狀態。成功的復原將產生一個事件：「資料庫叢集處於無法升級的狀態：DocumentDB 叢集處於無法順利完成主要版本升級的狀態。」此時，您應該聯繫 AWS 支持團隊進行故障排除並重新嘗試版本升級。您可以像以前一樣繼續使用工作負載。在任何其他罕見的情況下，升級所花費的時間超過預期，請聯絡 AWS 支援團隊尋求協助。

Amazon DocumentDB 3.6/4.0 到 5.0 升級的集群和新的 Amazon DocumentDB 5.0 集群之間的差異

- 多個數字數據類型的子文檔比較：
 - 如果叢集是從 Amazon DocumentDB 3.6 遷移，它將繼承 Amazon DocumentDB 3.6 子文檔比較行為。功能差異僅限於子文檔中的數字類型（例如長，雙精度，小數 128）。例如，`{a: {b: {NumberLong(1)}}` 不 `{a: {b: 1}}` 在 Amazon DocumentDB 3.6 相等，而它們被比較為在 Amazon DocumentDB 4.0 和之後相等。
 - 此子文件比較行為僅存在於 Amazon DocumentDB 3.6 和使用就地主要版本升級從 3.6 版升級的 Amazon DocumentDB 5.0 叢集中。這不適用於新建立的 Amazon DocumentDB 5.0 叢集。
- 就地主要版本升級會在升級的叢集上保留原始索引。我們建議您重新建立索引，以充分利用 Amazon DocumentDB 5.0 特定的效能改進（例如記憶體回收）。不過，重新建立索引可能需要額外的 I/O 和時間。如需詳細資訊，請參閱 [管理 Amazon DocumentDB 索引](#)。

Note

有關 Amazon DocumentDB 3.6/4.0 和 Amazon DocumentDB 5.0 之間的功能差異列表，請參閱 [MongoDB 性](#)

Amazon DocumentDB

雲端安全是 AWS 最重視的一環。身為 AWS 的客戶，您將能從資料中心和網路架構中獲益，這些都是專為最重視安全的組織而設計的。

安全是 AWS 與您共同肩負的責任。本文件有助於您了解如何在使用 Amazon DocumentDB 時套用共同責任模型。[共同的責任模型](#) 將此描述為雲端本身的安全和雲端內部的安全：

- 雲端本身的安全：AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。在 [AWS 合規計劃](#) 中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用於 Amazon DocumentDB (and MongoDB 兼容性)，請參閱 [合規計劃範圍內的 AWS 服務](#)。
- 雲端內部的安全：您的責任取決於所使用的 AWS 服務。您也必須對資料敏感度、組織要求，以及適用法律和法規等其他因素負責。

Note

本章適用於以執行個體為基礎的叢集和彈性叢集。如需詳細資訊，請參閱下列主題。

您也會了解如何使用其他 AWS 服務來協助您監控並保護 Amazon DocumentDB。下列主題說明如何將 Amazon DocumentDB 設定為符合您的安全與合規目標。

主題

- [Amazon DocumentDB 中的數據保護](#)
- [Amazon DocumentDB 的 Identity and Access Management](#)
- [Amazon DocumentDB 資料庫](#)
- [使用角色型存取控制存取資料庫](#)
- [在 Amazon DocumentDB 中進行記錄和監控控制](#)
- [更新您的 Amazon DocumentDB TLS 證書](#)
- [更新您的 Amazon DocumentDB TLS 證書- GovCloud \(美國西部\)](#)
- [Amazon DocumentDB 中的合規驗證](#)
- [Amazon DocumentDB ront 中的恢復能力](#)
- [亞馬遜文檔數據庫的基礎設施安全](#)
- [Amazon DocumentDB 的安全最佳實務](#)

- [稽核 Amazon DocumentDB 事件](#)

Amazon DocumentDB 中的數據保護

AWS [共用責任模型](#)適用於中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端的全球基礎設施。您負責維護在此基礎設施上託管內容的控制權。您也必須負責您所使用 AWS 服務的安全組態和管理任務。如需有關資料隱私權的更多相關資訊，請參閱[資料隱私權常見問答集](#)。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 [AWS 共同的責任模型和 GDPR](#) 部落格文章。

基於資料保護目的，建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶憑證，並設定個人使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 AWS CloudTrail 設定 API 和使用者活動日誌記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務內的所有預設安全控制項。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱[聯邦資訊處理標準 \(FIPS\) 140-2 概觀](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如 Name (名稱) 欄位。這包括當您使用主控台、API 或開發套件AWS 服務使用 Amazon 文件資料庫或AWS其他時間。AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

主題

- [用戶端欄位層級加密](#)
- [加密 Amazon DocumentDB 中的靜態數據](#)
- [對傳輸中資料進行加密](#)
- [金鑰管理](#)

用戶端欄位層級加密

Amazon DocumentDB 用戶端欄位層級加密 (FLE) 可讓您先加密用戶端應用程式中的敏感資料，然後再將其傳輸到 Amazon DocumentDB 叢集。在叢集中儲存和處理敏感資料時，機密資料會保持加密狀態，擷取時會在用戶端應用程式中解密。

主題

- [入門](#)
- [在用戶端 FLE 中查詢](#)
- [限制](#)

入門

Amazon DocumentDB 中用戶端 FLE 的初始組態是四個步驟的程序，包括建立加密金鑰、將角色與應用程式相關聯、設定應用程式，以及使用加密選項定義 CRUD 作業。

主題

- [步驟 1：建立加密金鑰](#)
- [步驟 2：建立角色與應用程式的關聯](#)
- [步驟 3：設定此應用程式](#)
- [步驟 4：定義 CRUD 作業](#)
- [範例：用戶端欄位層級加密設定檔](#)

步驟 1：建立加密金鑰

使用時AWS Key Management Service，建立用於加密和解密敏感資料欄位的對稱金鑰，並提供必要的IAM 使用許可。AWS KMS儲存用於加密資料金鑰 (DK) 的客戶金鑰 (CK)。我們建議您將客戶金鑰儲存在 KMS 中，以加強您的安全性狀態。資料金鑰是儲存在 Amazon DocumentDB 集合中的次要金鑰，必須先加密敏感欄位，才能將文件儲存在 Amazon DocumentDB 中。客戶金鑰會加密資料金鑰，進而加密和解密您的資料。如果您使用全域叢集，您可以建立多區域金鑰，供不同區域中的不同服務角色使用。

如需有關的詳細資訊AWS Key Management Service，包括如何建立金鑰，請參閱金[AWS鑰管理服務開發人員指南](#)。

步驟 2：建立角色與應用程式的關聯

建立具有適當AWS KMS許可的 IAM 政策。此政策允許其連接的 IAM 身分加密和解密資源欄位中指定的 KMS 金鑰。您的應用程式會假設此 IAM 角色進行驗證AWS KMS。

政策看起來會與此類似：

```
{ "Effect": "Allow",
  "Action": ["kms:Decrypt", "kms:Encrypt"],
  "Resource": "Customer Key ARN"
}
```

步驟 3：設定此應用程式

到目前為止，您已在其中定義了客戶金鑰AWS KMS並建立 IAM 角色，並為其提供正確的 IAM 許可以存取客戶金鑰。匯入必要套件。

```
import boto3
import json
import base64
from pymongo import MongoClient
from pymongo.encryption import (Algorithm,
                               ClientEncryption)
```

```
# create a session object:
my_session = boto3.session.Session()

# get access_key and secret_key programmatically using get_frozen_credentials() method:
current_credentials = my_session.get_credentials().get_frozen_credentials()
```

1. 指定 'aws' 作為 KMS 提供者類型，並輸入在上一步中擷取的帳戶登入資料。

```
provider = "aws"
kms_providers = {
    provider: {
        "accessKeyId": current_credentials.access_key,
        "secretAccessKey": current_credentials.secret_key
    }
}
```

2. 指定用於加密資料金鑰的客戶金鑰：

```
customer_key = {
  "region": "AWS region of the customer_key",
  "key": "customer_key ARN"
}

key_vault_namespace = "encryption.dataKeys"

key_alt_name = 'TEST_DATA_KEY'
```

3. 配置對 MongoClient 象：

```
client = MongoClient(connection_string)

coll = client.test.coll
coll.drop()

client_encryption = ClientEncryption(
  kms_providers, # pass in the kms_providers variable from the previous step
  key_vault_namespace = key_vault_namespace,
  client,
  coll.codec_options
)
```

4. 生成您的數據密鑰：

```
data_key_id = client_encryption.create_data_key(provider,
  customer_key,
  key_alt_name = [key_alt_name])
```

5. 擷取您現有的資料金鑰：

```
data_key = DataKey("aws",
  master_key = customer_key)
key_id = data_key["_id"]
data_key_id = client[key_vault_namespace].find_one({"_id": key_id})
```

步驟 4：定義 CRUD 作業

使用加密選項定義 CRUD 操作。

1. 定義集合寫/讀/刪除單個文檔：


```
coll = client.gameinfo.users
```

2. 顯式加密-加密欄位並插入：

Note

必須提供「key_id」或「key_alt_name」中的一個。

```
encrypted_first_name = client_encryption.encrypt(
    "Jane",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)
encrypted_last_name = client_encryption.encrypt(
    "Doe",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)
encrypted_dob = client_encryption.encrypt(
    "1990-01-01",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Random,
    key_alt_name=data_key_id
)

coll.insert_one(
    {"gamerTag": "jane_doe90",
     "firstName": encrypted_first_name,
     "lastName": encrypted_last_name,
     "dateOfBirth": encrypted_dob,
     "Favorite_games":["Halo", "Age of Empires 2", "Medal of Honor"]}
})
```

範例：用戶端欄位層級加密設定檔

在下列範例中，將每個#####取代為您自己的資訊。

```
# import python packages:
import boto3
import json
```

```
import base64
from pymongo import MongoClient
from pymongo.encryption import (Algorithm,
                                ClientEncryption)

def main():

    # create a session object:
    my_session = boto3.session.Session()

    # get aws_region from session object:
    aws_region = my_session.region_name

    # get access_key and secret_key programmatically using get_frozen_credentials()
    method:
    current_credentials = my_session.get_credentials().get_frozen_credentials()
    provider = "aws"

    # define the kms_providers which is later used to create the Data Key:
    kms_providers = {
        provider: {
            "accessKeyId": current_credentials.access_key,
            "secretAccessKey": current_credentials.secret_key
        }
    }

    # enter the kms key ARN. Replace the example ARN value.
    kms_arn = "arn:aws:kms:us-east-1:123456789:key/abcd-efgh-ijkl-mnop"
    customer_key = {
        "region": aws_region,
        "key": kms_arn
    }

    # secrets manager is used to store and retrieve user credentials for connecting to
    an Amazon DocumentDB cluster.
    # retrieve the secret using the secret name. Replace the example secret key.
    secret_name = "/dev/secretKey"
    docdb_credentials = json.loads(my_session.client(service_name = 'secretsmanager',
    region_name = "us-east-1").get_secret_value(SecretId = secret_name)['SecretString'])

    connection_params = '/?tls=true&tlsCAFile=global-
    bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
```

```
conn_str = 'mongodb://' + docdb_credentials["username"] + ':' +
docdb_credentials["password"] + '@' + docdb_credentials["host"] + ':' +
str(docdb_credentials["port"]) + connection_params
client = MongoClient(conn_str)

coll = client.test.coll
coll.drop()

# store the encryption data keys in a key vault collection (having naming
convention as db.collection):
key_vault_namespace = "encryption.dataKeys"
key_vault_db_name, key_vault_coll_name = key_vault_namespace.split(".", 1)

# set up the key vault (key_vault_namespace) for this example:
key_vault = client[key_vault_db_name][key_vault_coll_name]
key_vault.drop()
key_vault.create_index("keyAltNames", unique=True)

client_encryption = ClientEncryption(
    kms_providers,
    key_vault_namespace,
    client,
    coll.codec_options)

# create a new data key for the encrypted field:
data_key_id = client_encryption.create_data_key(provider, master_key=customer_key,
key_alt_names=["some_key_alt_name"], key_material = None)

# explicitly encrypt a field:
encrypted_first_name = client_encryption.encrypt(
    "Jane",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_id=data_key_id
)
coll.insert_one(
    {"gamerTag": "jane_doe90",
    "firstName": encrypted_first_name
})
doc = coll.find_one()
print('Encrypted document: %s' % (doc,))

# explicitly decrypt the field:
doc["encryptedField"] = client_encryption.decrypt(doc["encryptedField"])
print('Decrypted document: %s' % (doc,))
```

```
# cleanup resources:
client_encryption.close()
client.close()

if __name__ == "__main__":
    main()
```

在用戶端 FLE 中查詢

Amazon DocumentDB 支援使用用戶端 FLE 的點相等查詢。不平等和比較查詢可能會傳回不正確的結果。與對解密值發出相同的操作相比，讀取和寫入操作可能具有意外或不正確的行為。

例如，若要查詢玩家分數大於 500 的文件篩選條件：

```
db.users.find( {
  "gamerscore" : { $gt : 500 }
})
```

用戶端會使用明確加密方法來加密查詢值：

```
encrypted_gamerscore_filter = client_encryption.encrypt(
    500,
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)

db.users.find( {
  "gamerscore" : { $gt : encrypted_gamerscore_filter }
} )
```

在尋找作業中，Amazon DocumentDB 會使用大於不等式檢查，將加密值 500 與儲存在每個文件中的加密欄位值進行比較。使用解密的資料和值執行時，find 作業中的不等式檢查可能會傳回不同的結果，即使作業成功產生結果也是如此。

限制

下列限制適用於 Amazon DocumentDB 用戶端欄位層級加密：

- Amazon DocumentDB 僅支持點相等查詢。不平等和比較查詢可能會傳回不正確的結果。與對解密值發出相同的操作相比，讀取和寫入操作可能具有意外或不正確的行為。若要查詢玩家分數大於 500 的文件篩選器。

```
db.users.find( {  
  "gamerscore" : { $gt : 500 }  
})
```

用戶端使用明確加密方法來加密查詢值。

```
encrypted_gamerscore_filter = client_encryption.encrypt(  
  500,  
  Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,  
  key_alt_name=data_key_id  
)  
  
db.users.find({  
  "gamerscore" : { $gt : encrypted_gamerscore_filter }  
})
```

在尋找作業中，Amazon DocumentDB 會使用大於不等式檢查，將加密值 500 與儲存在每個文件中的加密欄位值進行比較。使用解密的資料和值執行時，find 作業中的不等式檢查可能會傳回不同的結果，即使作業成功產生結果也是如此。

- Amazon DocumentDB 不支持從蒙戈殼牌顯式客戶端 FLE。但是，該功能適用於我們支持的任何驅動程序。

加密 Amazon DocumentDB 中的靜態數據

Note

AWS KMS 正在將術語客戶主金鑰 (CMK) 取代為 AWS KMS key 和 KMS 金鑰。概念並沒有變更。為了防止重大變更，AWS KMS 會保留此術語的一些變化。

您可以在建立叢集時指定儲存加密選項，來加密 Amazon DocumentDB 叢集中的靜態資料。儲存加密功能一經啟用，範圍擴及整個叢集，並套用到所有執行個體，包括主執行個體和任何複本。它也會套用到叢集的儲存磁碟區、資料、索引、日誌、自動備份和快照。

Amazon DocumentDB 使用 256 位元進階加密標準 (AES-256)，使用儲存在中的加密金鑰，加密您的資料。使用已啟用靜態加密的 Amazon DocumentDB 叢集

時，不需要修改應用程式邏輯或用戶端連線。Amazon DocumentDB 會以透明的方式處理您的資料加密和解密，同時盡量降低對效能的影響。

Amazon DocumentDB 整合AWS KMS並使用稱為信封加密的方法來保護您的資料。使用加密 Amazon [文件資料庫叢集時AWS KMS](#)，Amazon DocumentDB 會AWS KMS要求使用您的 KMS 金鑰產生加密資料金鑰來加密儲存磁碟區。加密文字資料金鑰會使用您定義的 KMS 金鑰加密，並與加密的資料和儲存中繼資料一起儲存。當 Amazon DocumentDB 需要存取您的加密資料時，它會請求AWS KMS使用 KMS 金鑰解密密文資料金鑰，並在記憶體中快取純文字資料金鑰，以便有效率地加密和解密儲存磁碟區中的資料。

Amazon DocumentDB 中的儲存加密功能適用於所有支援的執行個體大小，以及所有AWS 區域提供 Amazon DocumentDB 的執行個體大小。

為 Amazon 文件資料庫叢集啟用靜態加密

當使用或AWS Command Line Interface (AWS CLI) 佈建叢集時，您可以在 Amazon DocumentDB 叢集上啟用AWS Management Console或停用靜態加密。您使用主控台所建立的叢集，依預設會啟用靜態加密。您使用 AWS CLI 所建立的叢集，依預設會停用靜態加密。因此，您必須使用 `--storage-encrypted` 參數明確啟用靜態加密。在任何一種情況下，當叢集建立之後，您都無法變更靜態加密選項。

Amazon DocumentDB 用AWS KMS於擷取和管理加密金鑰，以及定義控制這些金鑰使用方式的策略。如果您未指定金AWS KMS鑰識別碼，Amazon DocumentDB 會使用預設的AWS受管服務 KMS 金鑰。Amazon DocumentDB 為您AWS 區域的AWS 帳戶。如需詳細資訊，請參閱 [AWS Key Management Service 概念](#)。

若要開始建立自己的 KMS 金鑰，請參閱[開AWS Key Management Service](#)發人員指南中的入門指南。

Important

您必須使用對稱加密 KMS 金鑰來加密叢集，因為 Amazon DocumentDB 僅支援對稱加密 KMS 金鑰。請勿使用非對稱 KMS 金鑰來嘗試加密 Amazon DocumentDB 叢集中的資料。如需詳細資訊，請參閱AWS Key Management Service開發人員指南AWS KMS中的[非對稱金鑰](#)。

如果 Amazon DocumentDB 再也無法存取叢集加密金鑰 — 例如，當撤銷存取金鑰時 — 加密的叢集會進入終止狀態。在此情況下，您只能從備份中還原叢集。對於 Amazon DocumentDB，備份始終啟用 1 天。

此外，如果停用加密 Amazon DocumentDB 叢集的金鑰，您最終將失去該叢集的讀取和寫入存取權。當 Amazon DocumentDB 遇到經過金鑰加密的叢集，但它無法存取該金鑰時，叢集會使叢集進入終止狀態。在此情況下，該叢集再也無法使用，而且無法復原資料庫的目前狀態。若要還原叢集，您必須重新啟用 Amazon DocumentDB 加密金鑰的存取，然後從備份中還原叢集。

Important

在建立了加密叢集之後，您就無法變更它的 KMS 金鑰。務必在加密叢集建立之前決定您的加密金鑰需求。

Using the AWS Management Console

您可以在建立叢集時指定靜態加密選項。當您使用 AWS Management Console 建立叢集時，依預設會啟用靜態加密。叢集建立之後就無法變更此選項。

在建立叢集時指定靜態加密選項

1. 按照[入門](#)部分中所述建立 Amazon 文件資料庫叢集。不過，在步驟 6 中，請勿選擇 Create cluster (建立叢集)。
2. 在 Authentication (身分驗證) 區段下，選擇 Show advanced settings (顯示進階設定)。
3. 向下捲動至 Encryption-at-rest 區段。
4. 選擇您要靜態加密採用的選項。無論選擇哪個選項，叢集建立之後都無法變更。
 - 若要對此叢集中的資料啟用靜態加密，請選擇 Enable encryption (啟用加密)。
 - 如果您不要對此叢集中的資料啟用靜態加密，請選擇 Disable encryption (停用加密)。
5. 選擇您想要的主要金鑰。Amazon DocumentDB 使用 AWS Key Management Service (AWS KMS) 擷取和管理加密金鑰，以及定義控制這些金鑰使用方式的策略。如果您未指定金鑰識別碼，Amazon DocumentDB 會使用預設的 AWS 受管服務 KMS 金鑰。如需詳細資訊，請參閱 [AWS Key Management Service 概念](#)。

Note

建立了加密叢集之後，您就無法變更該叢集的金鑰。務必在加密叢集建立之前決定您的加密金鑰需求。

6. 請視需要完成其他區段，並建立您的叢集。

Using the AWS CLI

若要使用加密 Amazon DocumentDB 叢集AWS CLI，您必須在建立叢集時指定`--storage-encrypted`選項。使用建立的 Amazon DocumentDB 叢集預設AWS CLI不啟用儲存加密。

以下範例會建立 Amazon DocumentDB 叢集，同時啟用儲存加密。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --port 27017 \  
  --engine docdb \  
  --master-username yourMasterUsername \  
  --master-user-password yourMasterPassword \  
  --storage-encrypted
```

針對 Windows：

```
aws docdb create-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --port 27017 ^  
  --engine docdb ^  
  --master-username yourMasterUsername ^  
  --master-user-password yourMasterPassword ^  
  --storage-encrypted
```

建立加密的 Amazon DocumentDB 叢集時，您可以指定AWS KMS金鑰識別碼，如下列範例所示。


Example

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --port 27017 \  
  --engine docdb \  
  --master-username yourMasterUsername \  
  --master-user-password yourMasterPassword \  
  --storage-encrypted \  
  --kms-key-id key-arn-or-alias
```


針對 Windows :

```
aws docdb create-db-cluster ^
  --db-cluster-identifier sample-cluster ^
  --port 27017 ^
  --engine docdb ^
  --master-username yourMasterUsername ^
  --master-user-password yourMasterPassword ^
  --storage-encrypted ^
  --kms-key-id key-arn-or-alias
```

 Note

建立了加密叢集之後，您就無法變更該叢集的 KMS 金鑰。務必在加密叢集建立之前決定您的加密金鑰需求。

Amazon DocumentDB 加密叢集的限制

Amazon DocumentDB 加密叢集具有下列限制。

- 您只能在建立 Amazon DocumentDB 叢集時啟用或停用靜態加密，而不能在建立叢集之後啟用或停用該叢集的靜態加密。不過，您可以建立未加密叢集的快照，以建立未加密叢集的加密副本，然後在指定靜態加密選項的同時將未加密的快照還原為新叢集。

如需詳細資訊，請參閱下列主題：

- [建立手動叢集快照](#)
- [從叢集快照還原](#)
- [複製 Amazon DocumentDB 叢集快照](#)
- 已啟用儲存加密的 Amazon DocumentDB 叢集無法修改為停用加密。
- Amazon DocumentDB 叢集中的所有執行個體、自動備份、快照和索引都會使用相同的 KMS 金鑰加密。

對傳輸中資料進行加密

您可以使用傳輸層安全性 (TLS) 來加密應用程式和 Amazon DocumentDB 叢集之間的連線。根據預設，新建立的 Amazon DocumentDB 叢集會啟用傳輸中加密。它可以選擇性地在建立叢集時或稍後停

用。當啟用傳輸中的加密時，需要使用 TLS 的安全連線來連線到叢集。如需使用 TLS 連接到 Amazon DocumentDB 的詳細資訊，請參閱[以編程方式連接 Amazon DocumentDB](#)。

管理 Amazon DocumentDB 叢集 TLS 設定

[Amazon DocumentDB 叢集的傳輸中加密是透過叢集參數群組中的 TLS 參數來管理](#)。您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 管理您的 Amazon DocumentDB 叢集 TLS 設定。請參閱下列章節，了解如何驗證和修改您目前的 TLS 設定。

Using the AWS Management Console

請遵循下列步驟，使用主控台執行 TLS 加密的管理工作，例如識別參數群組、驗證 TLS 值，以及進行必要的修改。

Note

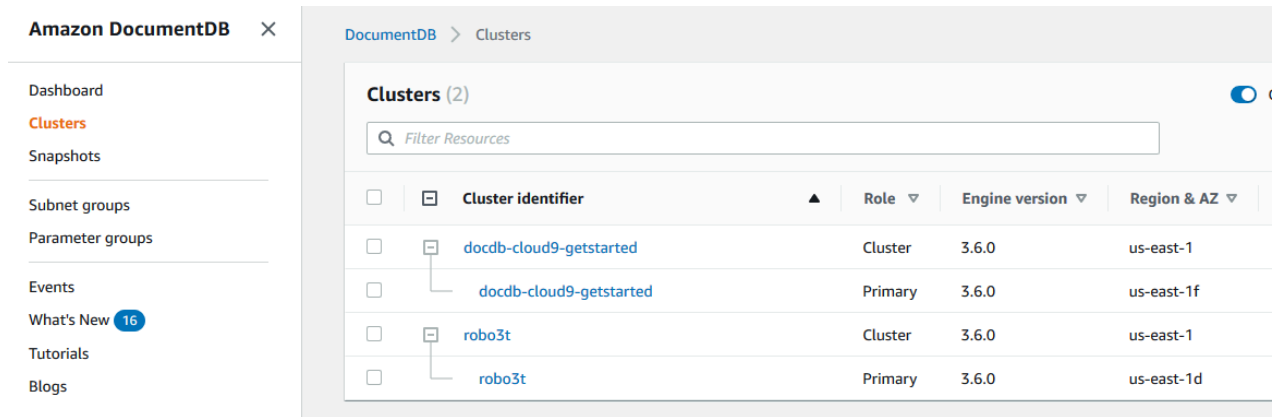
除非您在建立叢集時有不同指定，否則系統會使用預設的叢集參數群組來建立您的叢集。default 叢集參數群組中的參數無法修改 (例如，啟用/停用的 tls)。因此如果您的叢集使用 default 叢集參數群組，則您需要修改叢集，才能使用非預設叢集參數群組。首先，您可能需要建立自訂叢集參數群組。如需詳細資訊，請參閱[建立 Amazon DocumentDB 叢集參數群組](#)。

1. 判定您叢集正在使用的叢集參數群組。
 - a. [打開 Amazon DocumentDB 控制台 https://console.aws.amazon.com/docdb](https://console.aws.amazon.com/docdb)。
 - b. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

- c. 請注意，在「叢集」導覽方塊中，「叢集識別碼」欄會同時顯示叢集和執行個體。執行個體列在叢集下方。請參閱下面的屏幕截圖以供參考。



- d. 選擇您感興趣的叢集。
- e. 選擇組態索引標籤，然後向下捲動至叢集詳細資料底部，然後找出叢集參數群組。請記下叢集參數群組的名稱。

如果叢集參數群組的名稱為 default (例如，default.docdb3.6)，則您必須具有一個自訂叢集參數群組，並使其成為叢集的參數群組，然後再繼續。如需詳細資訊，請參閱下列內容：

1. [建立 Amazon DocumentDB 叢集參數群組](#)— 如果您沒有可以使用的自訂叢集參數群組，請建立一個群組。
2. [修改 Amazon DocumentDB 叢集](#)— 修改叢集以使用自訂叢集參數群組。

2. 判定 **tls** 叢集參數的目前值。
 - a. [打開 Amazon DocumentDB 控制台 https://console.aws.amazon.com/docdb](https://console.aws.amazon.com/docdb)。
 - b. 在導覽窗格中，選擇 Parameter groups (參數群組)。
 - c. 從叢集參數群組的清單中，選擇您有興趣之叢集參數群組的名稱。
 - d. 找出 Cluster parameters (叢集參數) 區段。在叢集參數清單中，找出 **tls** 叢集參數列。此時，必須留意下列四欄：
 - 叢集參數名稱 — 叢集參數的名稱。若為管理 TLS，您對 **tls** 叢集參數有興趣。
 - 值 — 每個叢集參數的目前值。
 - 允許的值 — 可套用至叢集參數的值清單。
 - 套用類型 — 靜態或動態。靜態叢集參數的變更必須在執行個體重新啟動後才能套用。可以立即或在執行個體重新啟動後套用動態叢集參數的變更。

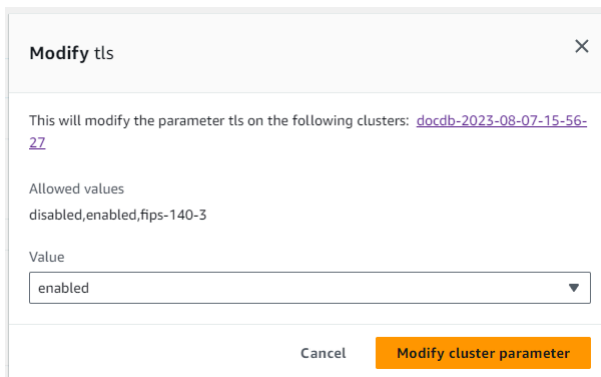
3. 修改 **tls** 叢集參數的值。

如果 `tls` 的值不是所需的值，請針對這個叢集參數群組修改其值。若要變更 `tls` 叢集參數的值，請按照以下步驟從上節繼續進行。

- a. 選擇叢集參數名稱 (`tls`) 左側的按鈕。
- b. 選擇編輯。
- c. 若要變更 `tls` 的值，請在 **Modify `tls`** (修改) 對話方塊中，從下拉式清單選擇您想要給與叢集參數的值。

有效的值如下：

- 停用 — 停用 TLS
- 已啟用 — 啟用 TLS (版本 1.0、1.1、1.2 和 1.3 版)
- FIPS-140-使用 FIPS 啟用 TLS。根據聯邦資訊處理標準 (FIPS) 出版物 140-3 的要求，叢集僅接受安全連線。只有從下列區域中的 Amazon DocumentDB 5.0 (引擎版本 3.0.3727) 叢集開始才支援此功能：ca-central-1、us-west-2、us-east-1、us-east-2、-1、-1。 us-gov-east us-gov-west



The screenshot shows a dialog box titled "Modify tls" with a close button (X) in the top right corner. The main text reads: "This will modify the parameter `tls` on the following clusters: [docdb-2023-08-07-15-56-22](#)". Below this, it lists "Allowed values: disabled, enabled, fips-140-3". There is a "Value" label above a dropdown menu that currently displays "enabled". At the bottom of the dialog, there are two buttons: "Cancel" and "Modify cluster parameter".

- d. 選擇 **Modify cluster parameter** (修改叢集參數)。此變更必須等到每個叢集執行個體重新啟動後才會套用。
4. 重新啟動 Amazon DocumentDB 實例。

重新啟動叢集的每個執行個體，讓變更套用到叢集中的所有執行個體。

- a. [打開 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>](https://console.aws.amazon.com/docdb)。
- b. 在導覽窗格中，選擇執行個體。
- c. 若要指定要重新啟動的執行個體，請在執行個體清單中找到執行個體，並選擇其名稱左側的按鈕。

- d. 選擇 Actions (動作)，然後 Reboot (重新啟動)。選擇 Reboot (重新啟動)，確認您要重新啟動。

Using the AWS CLI

請遵循下列步驟，使用 AWS CLI— 執行 TLS 加密的管理工作，例如識別參數群組、驗證 TLS 值，以及進行必要的修改。

Note

除非您在建立叢集時有不同指定，否則系統會使用預設的叢集參數群組來建立該叢集。default 叢集參數群組中的參數無法修改 (例如，啟用/停用的 tls)。因此如果您的叢集使用 default 叢集參數群組，則您需要修改叢集，才能使用非預設叢集參數群組。您可能需要先建立自訂叢集參數群組。如需詳細資訊，請參閱 [建立 Amazon DocumentDB 叢集參數群組](#)。

1. 判定您叢集正在使用的叢集參數群組。

使用 `describe-db-clusters` 命令搭配下列參數：

- **--db-cluster-identifier** – 必要。有興趣的叢集名稱。
- **--query** – 可選。將輸出僅限制為有興趣欄位的查詢，在此情況下，指的是叢集名稱及其叢集參數群組名稱。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier docdb-2019-05-07-13-57-08 \  
  --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  [  
    "docdb-2019-05-07-13-57-08",  
    "custom3-6-param-grp"  
  ]  
]
```

如果叢集參數群組的名稱為 `default` (例如, `default.docdb3.6`) , 則您必須具有一個自訂叢集參數群組, 並使其成為叢集的參數群組, 然後再繼續。如需詳細資訊, 請參閱下列主題:

1. [建立 Amazon DocumentDB 叢集參數群組](#)— 如果您沒有可以使用的自訂叢集參數群組, 請建立一個群組。
 2. [修改 Amazon DocumentDB 叢集](#)— 修改叢集以使用自訂叢集參數群組。
2. 判定 `tls` 叢集參數的目前值。

若要取得此叢集參數群組的詳細資訊, 請使用 `describe-db-cluster-parameters` 並搭配下列參數:

- `--db-cluster-parameter-group-name` – 必要。使用來自前一個命令輸出的叢集參數群組名稱。
- `--query` – 可選。將輸出僅限制為有興趣欄位的查詢, 在此情況下, 指的是 `ParameterName`、`ParameterValue`、`AllowedValues` 和 `ApplyType`。

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --query 'Parameters[*]'.  
[ParameterName,ParameterValue,AllowedValues,ApplyType]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  [  
    "audit_logs",  
    "disabled",  
    "enabled,disabled",  
    "dynamic"  
  ],  
  [  
    "tls",  
    "disabled",  
    "disabled,enabled,fips-140-3",  
    "static"  
  ],  
  [  
    "
```

```

        "ttl_monitor",
        "enabled",
        "disabled,enabled",
        "dynamic"
    ]
]

```

3. 修改 `tls` 叢集參數的值。

如果 `tls` 的值不是所需的值，請針對這個叢集參數群組修改其值。若要變更 `tls` 叢集參數的值，請使用 `modify-db-cluster-parameter-group` 操作搭配下列參數。

- **`--db-cluster-parameter-group-name`** – 必要。要修改之叢集參數群組的名稱。這不得為 `default.*` 叢集參數群組。
- **`--parameters`** – 必要。叢集參數群組之要修改的參數清單。
 - **`ParameterName`** – 必要。要修改之叢集參數的名稱。
 - **`ParameterValue`** – 必要。此叢集參數的新值。必須是其中一個叢集參數的 `AllowedValues`。
 - **`enabled`**— 叢集只接受使用 TLS 1.0、1.1、1.2 或 1.3 版的安全連線。
 - **`disabled`**— 叢集不接受使用 TLS 的安全連線。
 - **`fips-140-3`**— 叢集僅接受符合聯邦資訊處理標準 (FIPS) 出版物 140-3 的要求的安全連線。只有從下列區域中的 Amazon DocumentDB 5.0 (引擎版本 3.0.3727) 叢集開始才支援此功能：`ca-central-1`、`us-west-2`、`us-east-1`、`us-east-2`、`-1`、`-1`。 `us-gov-east` `us-gov-west`
 - **`ApplyMethod`**— 要套用此修改時。若為靜態叢集參數，例如 `tle`，此值必須是 `pending-reboot`。
 - **`pending-reboot`**— 變更只會在重新開機後套用至執行個體。您必須個別地重新啟動每個叢集執行個體，此變更才能在所有叢集執行個體中發生。

以下程式碼會停用 `tls`，並在每個資料庫執行個體重新啟動後套用。

```

aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name custom3-6-param-grp \
  --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"

```

下列程式碼會啟用 `tls` (版本 1.0、1.1. 1.2 和 1.3) , 並在重新啟動時將變更套用至每個資料庫執行個體。

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --parameters "ParameterName=tls,ParameterValue=enabled,ApplyMethod=pending-  
reboot"
```

下列程式碼會啟用 `TLSfips-140-3` , 並在重新啟動時將變更套用至每個資料庫執行個體。

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom5-0-param-grp \  
  --parameters  
  "ParameterName=tls,ParameterValue=fips-140-3,ApplyMethod=pending-reboot"
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

4. 重新啟動您的 Amazon DocumentDB 實例。

重新啟動叢集的每個執行個體，讓變更套用到叢集中的所有執行個體。若要重新啟動 Amazon DocumentDB 執行個體，請使用具有下列參數的 `reboot-db-instance` 作業：

- **`--db-instance-identifier`** – 必要。要重新啟動的執行個體識別碼。

以下程式碼會重新啟動執行個體 `sample-db-instance`。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb reboot-db-instance \  
  --db-instance-identifier sample-db-instance
```

針對 Windows：


```
aws docdb reboot-db-instance ^  
  --db-instance-identifier sample-db-instance
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBInstance": {  
    "AutoMinorVersionUpgrade": true,  
    "PubliclyAccessible": false,  
    "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",  
    "PendingModifiedValues": {},  
    "DBInstanceStatus": "rebooting",  
    "DBSubnetGroup": {  
      "Subnets": [  
        {  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-east-1a"  
          },  
          "SubnetIdentifier": "subnet-4e26d263"  
        },  
        {  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-east-1c"  
          },  
          "SubnetIdentifier": "subnet-afc329f4"  
        },  
        {  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-east-1e"  
          },  
          "SubnetIdentifier": "subnet-b3806e8f"  
        },  
        {  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-east-1d"  
          },  
          "SubnetIdentifier": "subnet-53ab3636"  
        },  
      ],  
    },  
  },  
}
```

```

        {
            "SubnetStatus": "Active",
            "SubnetAvailabilityZone": {
                "Name": "us-east-1b"
            },
            "SubnetIdentifier": "subnet-991cb8d0"
        },
        {
            "SubnetStatus": "Active",
            "SubnetAvailabilityZone": {
                "Name": "us-east-1f"
            },
            "SubnetIdentifier": "subnet-29ab1025"
        }
    ],
    "SubnetGroupStatus": "Complete",
    "DBSubnetGroupDescription": "default",
    "VpcId": "vpc-91280df6",
    "DBSubnetGroupName": "default"
},
"PromotionTier": 2,
"DBInstanceClass": "db.r5.4xlarge",
"InstanceCreateTime": "2018-11-05T23:10:49.905Z",
"PreferredBackupWindow": "00:00-00:30",
"KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-a50b-44d4-
b6a0-a177d5ff730b",
"StorageEncrypted": true,
"VpcSecurityGroups": [
    {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
    }
],
"EngineVersion": "3.6.0",
"DbiResourceId": "db-SAMPLERESOURCEID",
"DBInstanceIdentifier": "sample-cluster-instance-00",
"Engine": "docdb",
"AvailabilityZone": "us-east-1a",
"DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-cluster-
instance-00",
"BackupRetentionPeriod": 1,
"Endpoint": {
    "Address": "sample-cluster-instance-00.corcjzrlsfc.us-
east-1.docdb.amazonaws.com",

```

```
        "Port": 27017,  
        "HostedZoneId": "Z2R2ITUGPM61AM"  
    },  
    "DBClusterIdentifier": "sample-cluster"  
}  
}
```

執行個體重新啟動需要幾分鐘的時間來完成。您只能使用狀態為可用的執行個體。您可以使用主控台或 AWS CLI 監控執行個體的状态。如需更多詳細資訊，請參閱 [監控 Amazon DocumentDB 執行個體的状态](#)。

金鑰管理

Amazon DocumentDB 使用 AWS Key Management Service (AWS KMS) 來擷取和管理加密金鑰。AWS KMS 結合安全、高可用性的硬體和軟體，提供專為雲端擴充的金鑰管理系統。您可以使用 AWS KMS 建立加密金鑰，並定義控制這些金鑰使用方式的策略。AWS KMS 支援 AWS CloudTrail，因此您可稽核金鑰使用情況，以確認金鑰的使用方式是否正確。

您的 AWS KMS 金鑰可以與 Amazon DocumentDB 和支援的 AWS 服務結合使用，例如 Amazon Simple Storage Service (Amazon S3)、Amazon Relational Database Service 服務 (Amazon RDS)、Amazon Elastic Block Store (Amazon EBS) 和 Amazon Redshift。 [如需支援的服務清單 AWS KMS，請參閱 AWS Key Management Service 開發人員指南 AWS KMS 中的 AWS 服務使用方式](#)。如需 AWS KMS 的相關資訊，請參閱 [什麼是 AWS Key Management Service ?](#)

Amazon DocumentDB 的 Identity and Access Management

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。AWS 服務 IAM 管理員控制哪些人可以通過身份驗證 (登入) 和授權 (具有許可) 來使用 Amazon DocumentDB 資源。IAM 是您可以使用的 AWS 服務，無需額外付費。

主題

- [物件](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [Amazon DocumentDB 如何與 IAM 一起工作](#)
- [Amazon DocumentDB 的基於身份的政策示例](#)

- [疑難排解 Amazon DocumentDB 身分和存取](#)
- [管理您的 Amazon DocumentDB 資源的訪問許可](#)
- [對 Amazon DocumentDB 使用基於身份的政策 \(IAM 政策\)](#)
- [AWS Amazon DocumentDB 的受管政策](#)
- [Amazon DocumentDB API 許可：動作、資源和條件參考](#)

物件

您如何使用 AWS Identity and Access Management (IAM) 會有所不同，具體取決於您在 Amazon DocumentDB 中執行的工作。

服務使用者 — 如果您使用 Amazon DocumentDB 服務執行工作，則管理員會為您提供所需的登入資料和許可。當您使用更多 Amazon DocumentDB 功能來完成工作時，您可能需要額外的許可。瞭解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法在 Amazon DocumentDB 中訪問某個功能，請參閱。[疑難排解 Amazon DocumentDB 身分和存取](#)

服務管理員 — 如果您負責公司的 Amazon DocumentDB 資源，您可能可以完整存取 Amazon DocumentDB。您的任務是判斷服務使用者應存取哪些 Amazon DocumentDB 功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何將 IAM 與 Amazon DocumentDB 搭配使用，請參閱。[Amazon DocumentDB 如何與 IAM 一起工作](#)

IAM 管理員 — 如果您是 IAM 管理員，您可能想要了解如何撰寫政策以管理 Amazon DocumentDB 存取權的詳細資訊。若要檢視可在 IAM 中使用的 Amazon DocumentDB 以身分識別為基礎的政策範例，請參閱。[Amazon DocumentDB 的基於身份的政策示例](#)

使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》AWS 帳戶中的[如何登入](#)您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以便使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的[簽署 AWS API 請求](#)。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱《AWS IAM Identity Center 使用者指南》中的[多重要素驗證](#)和《IAM 使用者指南》中的[在 AWS 中使用多重要素驗證 \(MFA\)](#)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常作業。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱《IAM 使用者指南》中的[需要根使用者憑證的任務](#)。

聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時認證 AWS 服務 來存取。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identity Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。AWS Directory Service 同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的相關資訊，請參閱《AWS IAM Identity Center 使用者指南》中的[什麼是 IAM Identity Center ?](#)。

IAM 使用者和群組

[IAM 使用者](#)是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需詳細資訊，請參閱《IAM 使用者指南》<https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#rotate-credentials>中的為需要長期憑證的使用案例定期輪換存取金鑰。

[IAM 群組](#)是一種指定 IAM 使用者集合的身分。您無法以群組身分登入。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的過程變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱《IAM 使用者指南》中的[建立 IAM 使用者 \(而非角色\) 的時機](#)。

IAM 角色

[IAM 角色](#)是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以[切換角色，在中暫時擔任 IAM 角色](#)。AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的[使用 IAM 角色](#)。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並取得由角色定義的許可。如需有關聯合角色的詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-idp.html中的為第三方身分供應商建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱《AWS IAM Identity Center 使用者指南》中的[許可集](#)。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源 (而不是使用角色作為代理)。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的[IAM 角色與資源類型政策的差異](#)。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
- 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。當您使用某些服務時，您可能會執行一個動作，而該動作之後會在不同的服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱[《轉發存取工作階段》](#)。
- 服務角色：服務角色是服務擔任的 [IAM 角色](#)，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務 服務](#)。

- 服務連結角色 — 服務連結角色是連結至 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的 AWS 帳戶中，且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的[利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

如需了解是否要使用 IAM 角色或 IAM 使用者，請參閱《IAM 使用者指南》中的[建立 IAM 角色 \(而非使用者\) 的時機](#)。

使用政策管理存取權

您可以透過 AWS 建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS，當與身分識別或資源相關聯時，會定義其權限。AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱《IAM 使用者指南》中的[JSON 政策概觀](#)。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授與使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console、AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶。受管政策包括 AWS 受管政策和客戶管理

的策略。如需瞭解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的[在受管政策和內嵌政策間選擇](#)。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。AWS WAF 若要進一步了解 ACL，請參閱《Amazon Simple Storage Service 開發人員指南》中的[存取控制清單 \(ACL\) 概觀](#)。

其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授與您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 Principal 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱《IAM 使用者指南》中的[IAM 實體許可範圍](#)。
- 服務控制策略 (SCP) — SCP 是 JSON 策略，用於指定中組織或組織單位 (OU) 的最大權限。AWS Organizations 是一種用於分組和集中管理您企業擁有的多個 AWS 帳戶的服務。若您啟用組織中的所有功能，您可以將服務控制策略 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱《AWS Organizations 使用者指南》中的[SCP 運作方式](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作

階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱《IAM 使用者指南》中的[工作階段政策](#)。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的[政策評估邏輯](#)。

Amazon DocumentDB 如何與 IAM 一起工作

在您使用 IAM 管理 Amazon DocumentDB 的存取權限之前，請先了解哪些 IAM 功能可以搭配 Amazon DocumentDB 使用。

您可以與 Amazon DocumentDB 一起使用的 IAM 功能

| IAM 功能 | 執行個體型叢集 | 彈性叢集 |
|--------------------------------|---------|------|
| 身分型政策 | 是 | 是 |
| 資源型政策 | 否 | 否 |
| 政策動作 | 是 | 是 |
| 政策資源 | 是 | 是 |
| 政策條件索引鍵 (服務特定) | 是 | 是 |
| ACL | 否 | 否 |
| ABAC(政策中的標籤) | 部分 | 是 |
| 臨時憑證 | 是 | 是 |
| 主體許可 | 是 | 是 |
| 服務角色 | 是 | 是 |
| 服務連結角色 | 否 | 是 |

若要深入瞭解 Amazon DocumentDB 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱 IAM 使用者指南中的[的可與 IAM 搭配使用的 AWS 服務](#)。

Amazon DocumentDB 的基於身份的政策

支援身分型政策 是

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要瞭解如何建立身分類型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至附加的使用者或角色。如要瞭解您在 JSON 政策中使用的所有元素，請參閱 IAM 使用者指南中的[IAM JSON 政策元素參考](#)。

Amazon DocumentDB 的基於身份的政策示例

若要檢視 Amazon DocumentDB 以身分識別為基礎的政策範例，請參閱。[Amazon DocumentDB 的基於身份的政策示例](#)

Amazon DocumentDB 中基於資源的政策

支援以資源基礎的政策 否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或 AWS 服務

若要啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策附加到實體來授予許可。不過，如果資源型政策會為相同帳戶中的主體授與存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱 IAM 使用者指南中的[IAM 角色與資源型政策有何差異](#)。

Amazon DocumentDB 的政策操作

支援政策動作 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些操作需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授與執行相關聯操作的許可。

Note

對於某些管理功能，Amazon DocumentDB 使用與 Amazon Relational Database Service (Amazon RDS) 共享的操作技術。

若要查看 RDS 動作清單，請參閱[服務授權參考中由 Amazon Relational Database Service 務定義的動作](#)。

若要檢視 Amazon DocumentDB 元素叢集的政策動作，請參閱[服務授權參考中由 Amazon DocumentDB 彈性叢集定義的動作](#)。

Amazon DocumentDB 中的政策動作會在動作前使用下列前置詞：

```
aws
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [  
  "aws:action1",  
  "aws:action2"  
]
```

若要檢視 Amazon DocumentDB 以身分識別為基礎的政策範例，請參閱。[Amazon DocumentDB 的基於身份的政策示例](#)

Amazon DocumentDB 的政策資源

支援政策資源

是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出作業)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

Note

對於某些管理功能，Amazon DocumentDB 使用與 Amazon Relational Database Service (Amazon RDS) 共享的操作技術。

若要檢視 RDS 資源類型及其 ARN 的清單，請參閱服務授權參考中的 [Amazon 關聯式資料庫服務定義的資源](#)。若要了解可以使用哪些動作指定每個資源的 ARN，請參閱 [Amazon 關聯式資料庫服務定義的動作](#)。

若要檢視 Amazon DocumentDB 元素叢集的資源類型，請參閱服務授權參考中 [由 Amazon DocumentDB 彈性叢集定義的資源類型](#)。

若要檢視 Amazon DocumentDB 以身份識別為基礎的政策範例，請參閱 [Amazon DocumentDB 的基於身份的政策示例](#)

Amazon DocumentDB 的政策條件密鑰

支援服務特定政策條件索引鍵

是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用[條件運算子](#)的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯 OR 運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授與該 IAM 使用者。如需更多資訊，請參閱《IAM 使用者指南》中的[IAM 政策元素：變數和標籤](#)。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的[AWS 全域條件內容金鑰](#)。

Note

對於某些管理功能，Amazon DocumentDB 使用與 Amazon Relational Database Service (Amazon RDS) 共享的操作技術。

若要查看 RDS 條件金鑰清單，請參閱[服務授權參考中的 Amazon Relational Database Service 務的條件金鑰](#)。若要了解可以使用條件金鑰的動作和資源，請參閱[Amazon Relational Database Service 定義的動作](#)。

若要檢視 Amazon DocumentDB 元素叢集的條件金鑰，請參閱服務授權參考中的[Amazon DocumentDB 彈性叢集的條件金鑰](#)。

若要檢視 Amazon DocumentDB 以身分識別為基礎的政策範例，請參閱[Amazon DocumentDB 的基於身份的政策示例](#)

Amazon DocumentDB 中的 ACL

支援 ACL

否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 與 Amazon DocumentDB

Note

ABAC 僅部分支援執行個體型叢集，但彈性叢集支援。

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

若要根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的 [條件元素](#) 中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件索引鍵，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件索引鍵，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的 [什麼是 ABAC?](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的 [使用屬性型存取控制 \(ABAC\)](#)。

使用臨時登入資料與 Amazon DocumentDB

支援臨時憑證 是

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料 [搭配 AWS 服務 使用](#)，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱 IAM 使用者指南中的 [切換至角色 \(主控台\)](#)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。AWS 建議您動態產生臨時登入資料，而非使用長期存取金鑰。如需詳細資訊，請參閱 [IAM 中的暫時性安全憑證](#)。

Amazon DocumentDB 的跨服務主體許可

支援轉寄存取工作階段 (FAS) 是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。當您使用某些服務時，您可能會執行一個動作，而該動作之後會在不同的服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。

Amazon DocumentDB 的服務角色

支援服務角色 是

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的 [建立角色以委派許可給 AWS 服務 服務](#)。

Warning

變更服務角色的許可可能會中斷 Amazon DocumentDB 的功能。只有在 Amazon DocumentDB 提供指導時，才能編輯服務角色。

Amazon DocumentDB 的服務連結角色

Note

執行個體型叢集不支援服務連結角色，但彈性叢集支援。

服務連結角色是一種連結至 AWS 服務 服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服务連結角色的詳細資訊，請參閱 [可搭配 IAM 運作的 AWS 服務](#)。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

Amazon DocumentDB 的基於身份的政策示例

依預設，使用者和角色沒有建立或修改 Amazon DocumentDB 資源的權限。他們也無法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 來執行工作。若要授與使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

如需 Amazon DocumentDB 定義的動作和資源類型的詳細資訊，包括每個資源類型的 ARN 格式，請參閱服務授權參考中的[Amazon Relational Database Service 的動作、資源和條件金鑰](#)。

主題

- [政策最佳實務](#)
- [使用 Amazon DocumentDB 控制台](#)
- [允許使用者檢視他們自己的許可](#)

政策最佳實務

以身分識別為基礎的政策會決定某人是否可以在您的帳戶中建立、存取或刪除 Amazon DocumentDB 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始將權限授與使用者和工作負載，請使用可授與許多常見使用案例權限的 AWS 受管理原則。它們可用在您的 AWS 帳戶。建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的[AWS 受管政策](#)或[任務職能的 AWS 受管政策](#)。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的[IAM 中的政策和許可](#)。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 AWS CloudFormation。如需更多資訊，請參閱《IAM 使用者指南》中的[IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access

Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM Access Analyzer 政策驗證](#)。

- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 [IAM 使用者指南](#) 中的設定 MFA 保護的 API 存取。

有關 IAM 中最佳實務的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 最佳安全實務](#)。

使用 Amazon DocumentDB 控制台

若要存取 Amazon DocumentDB (具有 MongoDB 相容性) 主控台，您必須擁有一組最低限度的許可。這些許可必須允許您列 Amazon DocumentDB 和檢視有關 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

為了確保使用者和角色仍然可以使用 Amazon DocumentDB 主控台，請同時將 Amazon DocumentDB *ConsoleAccess* 或 *ReadOnly* AWS 受管政策附加到實體。如需詳細資訊，請參閱《IAM 使用者指南》中的 [新增許可到使用者](#)。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    }
  ]
}
```

```
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

疑難排解 Amazon DocumentDB 身分和存取

使用下列資訊來協助您診斷和修正使用 Amazon DocumentDB 和 IAM 時可能會遇到的常見問題。

主題

- [我沒有授權在 Amazon DocumentDB 中執行操作](#)
- [我沒有授權執行 iam : PassRole](#)
- [我想允許我以外的人訪問我 AWS 帳戶的 Amazon DocumentDB 資源](#)

我沒有授權在 Amazon DocumentDB 中執行操作

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 *aws:GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 `mateojackson` 使用者的政策，允許使用 `aws:GetWidget` 動作存取 `my-example-widget` 資源。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的登入憑證。

我沒有授權執行 `iam:PassRole`

如果您收到未獲授權執行 `iam:PassRole` 動作的錯誤訊息，則必須更新您的政策以允許您將角色傳遞給 Amazon DocumentDB。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM 使用者 `marymajor` 嘗試使用主控台在 Amazon DocumentDB 中執行動作時，會發生下列範例錯誤。但是，該動作要求服務具備服務角色授與的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的登入憑證。

我想允許我以外的人訪問我 AWS 帳戶 的 Amazon DocumentDB 資源

您可以建立一個角色，讓其他帳戶中的使用者或您的組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 Amazon DocumentDB 是否支援這些功能，請參閱 [Amazon DocumentDB 如何與 IAM 一起工作](#)
- 若要了解如何提供您所擁有資源 AWS 帳戶 的存取權，請參閱 [《IAM 使用者指南》中的另一個您擁有 AWS 帳戶 的 IAM 使用者提供存取權限](#)。
- 若要了解如何向第三方提供對資源的存取權 AWS 帳戶，請參閱 [IAM 使用者指南中的提供第三方 AWS 帳戶 擁有的存取權](#)。
- 如需了解如何透過聯合身分提供存取權，請參閱 [《IAM 使用者指南》中的將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。

- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 IAM 使用者指南中的 [IAM 角色與資源型政策的差異](#)。

管理您的 Amazon DocumentDB 資源的訪問許可

每個 AWS 資源都由一個擁有 AWS 帳戶，建立或存取資源的權限由權限原則控制。帳戶管理員可以將許可政策附加到 IAM 身分 (即使用者、群組和角色)，而某些服務 (例如 AWS Lambda) 也支援將權限政策附加至資源。

Note

帳戶管理員 (或管理員使用者) 是具有管理員許可的使用者。如需詳細資訊，請參《[IAM 使用者指南](#)》中的 IAM 最佳實務。

主題

- [Amazon DocumentDB 資源和操作](#)
- [了解資源所有權](#)
- [管理資源存取](#)
- [指定政策元素：動作、效果、資源和委託人](#)
- [在政策中指定條件](#)

Amazon DocumentDB 資源和操作

在 Amazon DocumentDB 中，主要資源是一個集群。Amazon DocumentDB 支援可與主要資源搭配使用的其他資源，例如執行個體、參數群組和事件訂閱。這些資源稱為「子資源」。

這些資源和子資源都有獨一無二的 Amazon Resource Name (ARN) 與其相關聯，如下表所示。

| 資源類型 | ARN 格式 |
|--------|---|
| 叢集 | arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-name</i> |
| 叢集參數群組 | arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i> |

| 資源類型 | ARN 格式 |
|-------|---|
| 叢集快照 | arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i> |
| 執行個體 | arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> |
| 安全群組 | arn:aws:rds: <i>region</i> : <i>account-id</i> :secgrp: <i>security-group-name</i> |
| 子網路群組 | arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> |

Amazon DocumentDB 提供了一組操作與 Amazon DocumentDB 資源的工作。如需可用操作的清單，請參閱[動作](#)。

了解資源所有權

資源擁有者 AWS 帳戶 是建立資源的人。也就是說，資源擁有者是驗證建立資源 AWS 帳戶 之請求的主體實體 (根帳戶、IAM 使用者或 IAM 角色) 的身分。下列範例說明其如何運作：

- 如果您使用的根帳戶登入資料 AWS 帳戶 來建立 Amazon DocumentDB 資源 (例如執行個體)，您 AWS 帳戶 就是 Amazon DocumentDB 資源的擁有者。
- 如果您在中建立 IAM 使用者，AWS 帳戶 並將建立 Amazon DocumentDB 資源的許可授與該使用者，則該使用者可以建立 Amazon DocumentDB 資源。不過 AWS 帳戶，使用者所屬的您擁有 Amazon DocumentDB 資源。
- 如果您在 AWS 帳戶 具有建立 Amazon DocumentDB 資源的許可中建立 IAM 角色，則任何可以擔任該角色的人都可以建立 Amazon DocumentDB 資源。您的 AWS 帳戶 角色所屬，擁有 Amazon DocumentDB 資源。

管理資源存取

許可政策描述誰可以存取哪些資源。下一節說明可用來建立許可政策的選項。

Note

本節討論在 Amazon DocumentDB 的內容中使用 IAM。它不提供 IAM 服務的詳細資訊。如需完整的 IAM 文件，請參閱 IAM 使用者指南中的 [什麼是 IAM](#)。如需 IAM 政策語法和說明的相關資訊，請參閱 [IAM 使用者指南中的 AWS IAM 政策參考](#)。

連接至 IAM 身分的政策稱為身分識別型政策 (IAM 政策)。連接到資源的政策稱為「資源類型」政策。Amazon DocumentDB 僅支援以身分識別為基礎的政策 (IAM 政策)。

主題

- [身分類型政策 \(IAM 政策\)](#)
- [資源型政策](#)

身分類型政策 (IAM 政策)

您可以將政策連接到 IAM 身分。例如，您可以執行下列動作：

- 將許可政策附加到帳戶中的使用者或群組 — 帳戶管理員可以使用與特定使用者相關聯的許可政策授與該使用者建立 Amazon DocumentDB 資源 (例如執行個體) 的許可。
- 將許可政策連接至角色 (授予跨帳戶許可)：您可以將身分識別型許可政策連接至 IAM 角色，藉此授予跨帳戶許可。例如，系統管理員可以建立角色，將跨帳戶權限授與另一個 AWS 帳戶或 AWS 服務，如下所示：
 1. 帳戶 A 管理員建立 IAM 角色，並將許可政策連接到可授與帳戶 A 中資源許可的角色。
 2. 帳戶 A 管理員將信任政策連接至該角色，識別帳戶 B 做為可擔任該角的委託人。
 3. 然後，帳戶 B 管理員就可以將擔任角色的權限委派給帳戶 B 中的任何使用者。這樣做可讓帳戶 B 中的使用者建立或存取帳戶 A 中的資源。如果您想要授與 AWS 服務擔任角色的權限，則信任策略中的主參與者也可以是 AWS 服務主體。

如需使用 IAM 來委派許可的詳細資訊，請參閱《IAM 使用者指南》中的 [存取管理](#)。

以下是範例政策，允許具有 ID 123456789012 的使用者為您的執行個體建立執行個體 AWS 帳戶。新的執行個體還須使用選項群組，以及以 default 開頭的資料庫參數群組，而且它還須使用 default 子網路群組。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowCreateDBInstanceOnly",
    "Effect": "Allow",
    "Action": [
      "rds:CreateDBInstance"
    ],
    "Resource": [
      "arn:aws:rds*:123456789012:db:test*",
      "arn:aws:rds*:123456789012:pg:cluster-pg:default*",
      "arn:aws:rds*:123456789012:subgrp:default"
    ]
  }
]
```

如需將身分型政策與 Amazon DocumentDB 搭配使用的詳細資訊，請參閱 [對 Amazon DocumentDB 使用基於身分的政策 \(IAM 政策\)](#) 如需使用者、群組、角色和許可的詳細資訊，請參閱《IAM 使用者指南》中的 [身分 \(使用者、群組和角色\)](#)。

資源型政策

其他服務，例如 Amazon Simple Storage Service (Amazon S3)，也支援以資源為基礎的許可政策。例如，您可以將政策連接至 Simple Storage Service (Amazon S3) 儲存貯體，以管理該儲存貯體的存取許可。Amazon DocumentDB 不支援以資源為基礎的政策。

指定政策元素：動作、效果、資源和委託人

對於每個 Amazon DocumentDB 源 (請參閱 [Amazon DocumentDB 資源和操作](#))，服務會定義一組 API 操作。如需詳細資訊，請參閱 [動作](#)。為了授與這些 API 操作的許可，Amazon DocumentDB 會定義一組您可以在政策中指定的動作。執行一項 API 操作可能需要多個動作的許可。

以下是基本的政策元素：

- 資源 – 在政策中，您可以使用 Amazon Resource Name (ARN) 來識別要套用政策的資源。
- 動作：使用動作關鍵字識別您要允許或拒絕的資源操作。例如，`rds:DescribeDBInstances` 許可允許使用者執行 `DescribeDBInstances` 操作。
- 效果 - 您可以指定使用者要求特定動作時會有什麼效果；可為允許或拒絕。如果您未明確授予存取 (允許) 資源，則隱含地拒絕存取。您也可以明確拒絕資源存取，這樣做可確保使用者無法存取資源，即使不同政策授予存取也是一樣。

- 委託人：在以身分為基礎的政策 (IAM 政策) 中，政策所連接的使用者就是隱含委託人。對於資源型政策，您可以指定想要收到許可的使用者、帳戶、服務或其他實體 (僅適用於資源型政策)。Amazon DocumentDB 不支援以資源為基礎的政策。

如需進一步了解有關 IAM 政策語法和說明的詳細資訊，請參閱《IAM 使用者指南》中的 [AWS IAM 政策參考](#)。

如需顯示所有 Amazon DocumentDB API 動作及其適用資源的表格，請參閱 [Amazon DocumentDB API 許可：動作、資源和條件參考](#)

在政策中指定條件

當您授與許可時，您可以使用 IAM 政策語言指定政策生效時間的條件。例如，建議只在特定日期之後套用政策。如需使用政策語言指定條件的詳細資訊，請參閱 IAM 使用者指南中的 [條件](#)。

欲表示條件，您可以使用預先定義的條件金鑰。Amazon DocumentDB 沒有可在 IAM 政策中使用的服務特定內容金鑰。如需所有服務可用的全域條件內容金鑰清單，請參閱 IAM 使用者指南中的 [條件可用金鑰](#)。

對 Amazon DocumentDB 使用基於身份的政策 (IAM 政策)

Important

對於某些管理功能，Amazon DocumentDB 會使用與 Amazon RDS 共用的操作技術。Amazon DocumentDB 主控 AWS CLI 台和 API 呼叫會記錄為對 Amazon RDS API 的呼叫。我們建議您先檢閱介紹性主題，其中說明可用於管理 Amazon DocumentDB 資源存取權的基本概念和選項。如需詳細資訊，請參閱 [管理您的 Amazon DocumentDB 資源的訪問許可](#)。

這個主題提供以身分為基礎的政策範例，在該政策中帳戶管理員可以將許可政策連接至 IAM 身分 (即使用者、群組和角色)。

以下是 IAM 政策的範例。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateDBInstanceOnly",
```



```
    "Effect": "Allow",
    "Action": [
        "rds:CreateDBInstance"
    ],
    "Resource": [
        "arn:aws:rds*:123456789012:db:test*",
        "arn:aws:rds*:123456789012:pg:cluster-pg:default*",
        "arn:aws:rds*:123456789012:subgrp:default"
    ]
}
]
```

政策包括單一陳述式，指定 IAM 使用者的下列許可：

- 此政策允許 IAM 使用者使用 [CreateDbInstance](#) 動作建立執行個體 (這也適用於[create-db-instance](#) AWS CLI 作業和)。AWS Management Console
- Resource 元素指定使用者可對資源或搭配資源執行動作。您可以使用 Amazon Resource Name (ARN) 來指定資源。此 ARN 包括資源所屬的服務名稱 (rds)、(在此範例中*表示任何區域)、使用者帳號 (在此範例中123456789012為使用者 ID) 以及資源類型。AWS 區域

範例中的 Resource 元素對使用者的資源指定下列政策限制：

- 新執行個體的執行個體識別碼必須以 test 開頭 (例如，testCustomerData1、test-region2-data)。
- 新執行個體的參數群組必須以 default 開頭。
- 新執行個體的子網路群組必須是 default 子網路群組。

此政策不指定 Principal 元素，因為您不會在以身分為基礎的政策中，指定取得許可的主體。當您將政策連接至使用者時，這名使用者即為隱含主體。當您將許可政策連接至 IAM 角色，該角色的信任政策中所識別的委託人即取得許可。

如需顯示所有 Amazon DocumentDB API 操作及其適用資源的表格，請參閱。[Amazon DocumentDB API 許可：動作、資源和條件參考](#)

使用 Amazon DocumentDB 主控台所需的許可

若要讓使用者使用 Amazon DocumentDB 主控台，該使用者必須擁有一組最低限度的許可。這些許可允許使用者描述他們的 Amazon DocumentDB 資源，AWS 帳戶 並提供其他相關資訊，包括 Amazon EC2 安全性和網路資訊。

如果您建立比最基本必要許可更嚴格的 IAM 政策，則對於採取該 IAM 政策的使用者而言，主控台就無法如預期運作。為確保這些使用者仍然可以使用 Amazon DocumentDB 主控台，請同時將 AmazonDocDBConsoleFullAccess 受管政策附加到使用者，如中所述。[AWS Amazon DocumentDB 的受管政策](#)

您不需要為只對 AWS CLI 或 Amazon DocumentDB API 進行呼叫的使用者允許最低主控台許可。

客戶受管政策範例

在本節中，您可以找到授與各種 Amazon DocumentDB 動作許可的範例使用者政策。當您使用 Amazon DocumentDB API 動作、AWS 開發套件或 AWS CLI 當您使用主控台時，需要授予主控台特定的額外許可，這會在「[使用 Amazon DocumentDB 主控台所需的許可](#)」中予以討論。

對於某些管理功能，Amazon DocumentDB 會使用與 Amazon Relational Database Service 服務 (Amazon RDS) 和亞馬 Amazon Neptune 共用的操作技術。

Note

所有範例均使用美國東部 (維吉尼亞北部) 區域 (us-east-1)，並包含虛擬帳戶 ID。

範例

- [範例 1：允許使用者對任何 Amazon 文件資源執行任何描述動作](#)
- [範例 2：防止使用者刪除執行個體](#)
- [範例 3：除非已啟用儲存區加密，否則防止使用者建立叢集](#)

範例 1：允許使用者對任何 Amazon 文件資源執行任何描述動作

下列許可政策會授予使用者執行開頭為 Describe 之所有動作的許可。這些動作會顯示 Amazon DocumentDB 資源 (例如執行個體) 的相關資訊。元素中的萬用字 Resource 元 (*) 表示該帳戶擁有的所有 Amazon DocumentDB 資源都允許執行這些動作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRDSDescribe",
      "Effect": "Allow",
      "Action": "rds:Describe*",
```

```
        "Resource": "*"
    }
]
}
```

範例 2：防止使用者刪除執行個體

以下許可政策授予許可，以防止使用者刪除特定的執行個體。例如，您可能想拒絕給予任何非管理員的使用者刪除生產執行個體的能力。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyDelete1",
      "Effect": "Deny",
      "Action": "rds:DeleteDBInstance",
      "Resource": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
    }
  ]
}
```

範例 3：除非已啟用儲存區加密，否則防止使用者建立叢集

除非啟用儲存加密，否則下列許可政策會拒絕使用者建立 Amazon DocumentDB 叢集的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventUnencryptedDocumentDB",
      "Effect": "Deny",
      "Action": "RDS:CreateDBCluster",
      "Condition": {
        "Bool": {
          "rds:StorageEncrypted": "false"
        }
      },
      "StringEquals": {
        "rds:DatabaseEngine": "docdb"
      }
    },
    {
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

AWS Amazon DocumentDB 的受管政策

若要新增使用者、群組和角色的權限，使用 AWS 受管理的原則比自己撰寫原則更容易。建立 [IAM 客戶受管政策](#) 需要時間和專業知識，而受管政策可為您的團隊提供其所需的許可。若要快速開始使用，您可以使用我們的 AWS 受管政策。這些政策涵蓋常見使用案例，並可在您的 AWS 帳戶中使用。如需有關 AWS 受管理策略的詳細資訊，請參閱 [AWS Identity and Access Management 使用指南](#) 中的受管理策略。

AWS 服務會維護和更新 AWS 受管理的策略。您無法變更 AWS 受管理原則中的權限。服務有時會將其他權限新增至受 AWS 管理的策略，以支援新功能。此類型的更新會影響已連接政策的所有身分識別 (使用者、群組和角色)。當新功能啟動或新作業可用時，服務最有可能更新 AWS 受管理的策略。服務不會從 AWS 受管理的政策移除權限，因此政策更新不會破壞您現有的權限。

此外，還 AWS 支援跨多個服務之工作職能的受管理原則。例如，[ViewOnlyAccess](#) AWS 受管理的原則會提供許多 AWS 服務和資源的唯讀存取權。當服務啟動新功能時，會為新作業和資源新 AWS 增唯讀權限。如需工作功能原則的清單與說明，請參閱《[AWS Identity and Access Management 使用者指南](#)》中的工作職能受管理原則。

下列 AWS 受管政策 (您可以附加到帳戶中的使用者) 專屬於 Amazon DocumentDB：

- [AmazonDoc資料庫 FullAccess](#)— 授予根 AWS 帳戶對所有 Amazon DocumentDB 源的完整存取權。
- [AmazonDoc資料庫 ReadOnlyAccess](#)— 授與根 AWS 帳戶的所有 Amazon DocumentDB 源的唯讀存取權。
- [AmazonDoc資料庫 ConsoleFullAccess](#)— 使用授予管理 Amazon DocumentDB 和 Amazon DocumentDB 彈性叢集資源的完整存取權。AWS Management Console
- [AmazonDoc資料庫 ElasticReadOnlyAccess](#)— 授與根 AWS 帳戶的所有 Amazon DocumentDB 彈性叢集資源的唯讀存取權。
- [AmazonDoc資料庫 ElasticFullAccess](#)— 授與根 AWS 帳戶的所有 Amazon DocumentDB 彈性叢集資源的完整存取權。

AmazonDoc資料庫 FullAccess

此政策授予管理許可，允許主體完整存取所有 Amazon DocumentDB 動作。此政策中的許可分組如下：

- Amazon DocumentDB 許可允許所有 Amazon DocumentDB 操作。
- 此政策中的某些 Amazon EC2 許可需要驗證 API 請求中傳遞的資源。這是為了確保 Amazon DocumentDB 能夠成功地將資源與叢集搭配使用。此政策中的其餘 Amazon EC2 許可允許 Amazon DocumentDB 建立所需的 AWS 資源，讓您能夠連線到叢集。
- API 呼叫期間會使用 Amazon DocumentDB 許可，以驗證請求中傳遞的資源。他們需要 Amazon DocumentDB 能夠使用傳遞的密鑰與 Amazon DocumentDB 集群。
- Amazon DocumentDB 需要 CloudWatch 日誌才能確保日誌傳遞目的地可連接，並且對於代理日誌使用有效。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "rds:AddRoleToDBCluster",
        "rds:AddSourceIdentifierToSubscription",
        "rds:AddTagsToResource",
        "rds:ApplyPendingMaintenanceAction",
        "rds:CopyDBClusterParameterGroup",
        "rds:CopyDBClusterSnapshot",
        "rds:CopyDBParameterGroup",
        "rds>CreateDBCluster",
        "rds>CreateDBClusterParameterGroup",
        "rds>CreateDBClusterSnapshot",
        "rds>CreateDBInstance",
        "rds>CreateDBParameterGroup",
        "rds>CreateDBSubnetGroup",
        "rds>CreateEventSubscription",
        "rds>DeleteDBCluster",
        "rds>DeleteDBClusterParameterGroup",
        "rds>DeleteDBClusterSnapshot",
        "rds>DeleteDBInstance",
        "rds>DeleteDBParameterGroup",
        "rds>DeleteDBSubnetGroup",
        "rds>DeleteEventSubscription",
        "rds:DescribeAccountAttributes",
        "rds:DescribeCertificates",
        "rds:DescribeDBClusterParameterGroups",
        "rds:DescribeDBClusterParameters",
        "rds:DescribeDBClusterSnapshotAttributes",
```

```

        "rds:DescribeDBClusterSnapshots",
        "rds:DescribeDBClusters",
        "rds:DescribeDBEngineVersions",
        "rds:DescribeDBInstances",
        "rds:DescribeDBLogFiles",
        "rds:DescribeDBParameterGroups",
        "rds:DescribeDBParameters",
        "rds:DescribeDBSecurityGroups",
        "rds:DescribeDBSubnetGroups",
        "rds:DescribeEngineDefaultClusterParameters",
        "rds:DescribeEngineDefaultParameters",
        "rds:DescribeEventCategories",
        "rds:DescribeEventSubscriptions",
        "rds:DescribeEvents",
        "rds:DescribeOptionGroups",
        "rds:DescribeOrderableDBInstanceOptions",
        "rds:DescribePendingMaintenanceActions",
        "rds:DescribeValidDBInstanceModifications",
        "rds:DownloadDBLogFilePortion",
        "rds:FailoverDBCluster",
        "rds:ListTagsForResource",
        "rds:ModifyDBCluster",
        "rds:ModifyDBClusterParameterGroup",
        "rds:ModifyDBClusterSnapshotAttribute",
        "rds:ModifyDBInstance",
        "rds:ModifyDBParameterGroup",
        "rds:ModifyDBSubnetGroup",
        "rds:ModifyEventSubscription",
        "rds:PromoteReadReplicaDBCluster",
        "rds:RebootDBInstance",
        "rds:RemoveRoleFromDBCluster",
        "rds:RemoveSourceIdentifierFromSubscription",
        "rds:RemoveTagsForResource",
        "rds:ResetDBClusterParameterGroup",
        "rds:ResetDBParameterGroup",
        "rds:RestoreDBClusterFromSnapshot",
        "rds:RestoreDBClusterToPointInTime"
    ],
    "Effect": "Allow",
    "Resource": [
        "*"
    ]
},
{

```

```

    "Action": [
      "cloudwatch:GetMetricStatistics",
      "cloudwatch:ListMetrics",
      "ec2:DescribeAccountAttributes",
      "ec2:DescribeAvailabilityZones",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeSubnets",
      "ec2:DescribeVpcAttribute",
      "ec2:DescribeVpcs",
      "kms:ListAliases",
      "kms:ListKeyPolicies",
      "kms:ListKeys",
      "kms:ListRetirableGrants",
      "logs:DescribeLogStreams",
      "logs:GetLogEvents",
      "sns:ListSubscriptions",
      "sns:ListTopics",
      "sns:Publish"
    ],
    "Effect": "Allow",
    "Resource": [
      "*"
    ]
  },
  {
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "rds.amazonaws.com"
      }
    }
  }
]
}

```

AmazonDoc資料庫 ReadOnlyAccess

此政策授予唯讀許可，允許使用者檢視 Amazon DocumentDB 中的資訊。附加此政策的主體無法進行任何更新或刪除現有資源，也無法建立新的 Amazon DocumentDB 資源。例如，具有這些許可的主體

可以檢視與其帳戶相關聯的叢集和組態清單，但無法變更任何叢集的組態或設定。此政策中的許可分組如下：

- Amazon DocumentDB 許可允許您列出 Amazon DocumentDB 資源，描述它們，並獲取有關它們的信息。
- Amazon EC2 許可用來描述與叢集關聯的 Amazon VPC、子網路、安全群組和 ENI。
- Amazon DocumentDB 權限是用來描述與叢集相關聯的金鑰。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "rds:DescribeAccountAttributes",
        "rds:DescribeCertificates",
        "rds:DescribeDBClusterParameterGroups",
        "rds:DescribeDBClusterParameters",
        "rds:DescribeDBClusterSnapshotAttributes",
        "rds:DescribeDBClusterSnapshots",
        "rds:DescribeDBClusters",
        "rds:DescribeDBEngineVersions",
        "rds:DescribeDBInstances",
        "rds:DescribeDBLogFiles",
        "rds:DescribeDBParameterGroups",
        "rds:DescribeDBParameters",
        "rds:DescribeDBSubnetGroups",
        "rds:DescribeEventCategories",
        "rds:DescribeEventSubscriptions",
        "rds:DescribeEvents",
        "rds:DescribeOrderableDBInstanceOptions",
        "rds:DescribePendingMaintenanceActions",
        "rds:DownloadDBLogFilePortion",
        "rds:ListTagsForResource"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics"
      ]
    }
  ]
}
```



```
    ],
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": [
      "ec2:DescribeAccountAttributes",
      "ec2:DescribeAvailabilityZones",
      "ec2:DescribeInternetGateways",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeSubnets",
      "ec2:DescribeVpcAttribute",
      "ec2:DescribeVpcs"
    ],
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": [
      "kms:ListKeys",
      "kms:ListRetirableGrants",
      "kms:ListAliases",
      "kms:ListKeyPolicies"
    ],
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": [
      "logs:DescribeLogStreams",
      "logs:GetLogEvents"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:logs:*:*:log-group:/aws/rds/*:log-stream:*",
      "arn:aws:logs:*:*:log-group:/aws/docdb/*:log-stream:*"
    ]
  }
]
```

AmazonDoc資料庫 ConsoleFullAccess

使用以下命令授予管理 Amazon 文件資源的完整存 AWS Management Console 取權：

- Amazon DocumentDB 許可允許所有 Amazon DocumentDB 和 Amazon DocumentDB 集群操作。
- 此政策中的某些 Amazon EC2 許可需要驗證 API 請求中傳遞的資源。這是為了確保 Amazon DocumentDB 能夠成功使用資源來佈建和維護叢集。此政策中的其餘 Amazon EC2 許可允許 Amazon DocumentDB 建立所需的 AWS 資源，讓您能夠連線到您的叢集，例如 vPSEndPoint。
- AWS KMS 在 API 調用期間使用權限 AWS KMS 來驗證請求中傳遞的資源。Amazon DocumentDB 需要它們才能使用傳遞的金鑰，透過 Amazon DocumentDB 彈性叢集來加密和解密靜態資料。
- Amazon DocumentDB 需要 CloudWatch 日誌才能確保日誌傳遞目的地可連接，並且對於稽核和分析日誌使用有效。
- 需要 Secrets Manager 員權限才能驗證指定的密碼，並使用它為 Amazon DocumentDB 彈性叢集設定管理員使用者。
- Amazon 文件資料庫叢集管理動作需要 Amazon RDS 許可。對於某些管理功能，Amazon DocumentDB 會使用與 Amazon RDS 共用的操作技術。
- SNS 許可允許主體使用亞馬遜簡單通知服務 (Amazon SNS) 訂閱和主題，以及發佈 Amazon SNS 訊息。
- 建立指標和記錄發佈所需的服務連結角色時，需要 IAM 許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DocdbSids",
      "Effect": "Allow",
      "Action": [
        "docdb-elastic:CreateCluster",
        "docdb-elastic:UpdateCluster",
        "docdb-elastic:GetCluster",
        "docdb-elastic>DeleteCluster",
        "docdb-elastic:ListClusters",
        "docdb-elastic:CreateClusterSnapshot",
        "docdb-elastic:GetClusterSnapshot",
        "docdb-elastic>DeleteClusterSnapshot",
        "docdb-elastic:ListClusterSnapshots",
        "docdb-elastic:RestoreClusterFromSnapshot",
        "docdb-elastic:TagResource",
```

```
"docdb-elastic:UntagResource",
"docdb-elastic:ListTagsForResource",
"docdb-elastic:CopyClusterSnapshot",
"docdb-elastic:StartCluster",
"docdb-elastic:StopCluster",
"rds:AddRoleToDBCluster",
"rds:AddSourceIdentifierToSubscription",
"rds:AddTagsToResource",
"rds:ApplyPendingMaintenanceAction",
"rds:CopyDBClusterParameterGroup",
"rds:CopyDBClusterSnapshot",
"rds:CopyDBParameterGroup",
"rds:CreateDBCluster",
"rds:CreateDBClusterParameterGroup",
"rds:CreateDBClusterSnapshot",
"rds:CreateDBInstance",
"rds:CreateDBParameterGroup",
"rds:CreateDBSubnetGroup",
"rds:CreateEventSubscription",
"rds:CreateGlobalCluster",
"rds>DeleteDBCluster",
"rds>DeleteDBClusterParameterGroup",
"rds>DeleteDBClusterSnapshot",
"rds>DeleteDBInstance",
"rds>DeleteDBParameterGroup",
"rds>DeleteDBSubnetGroup",
"rds>DeleteEventSubscription",
"rds>DeleteGlobalCluster",
"rds:DescribeAccountAttributes",
"rds:DescribeCertificates",
"rds:DescribeDBClusterParameterGroups",
"rds:DescribeDBClusterParameters",
"rds:DescribeDBClusterSnapshotAttributes",
"rds:DescribeDBClusterSnapshots",
"rds:DescribeDBClusters",
"rds:DescribeDBEngineVersions",
"rds:DescribeDBInstances",
"rds:DescribeDBLogFiles",
"rds:DescribeDBParameterGroups",
"rds:DescribeDBParameters",
"rds:DescribeDBSecurityGroups",
"rds:DescribeDBSubnetGroups",
"rds:DescribeEngineDefaultClusterParameters",
"rds:DescribeEngineDefaultParameters",
```

```

        "rds:DescribeEventCategories",
        "rds:DescribeEventSubscriptions",
        "rds:DescribeEvents",
        "rds:DescribeGlobalClusters",
        "rds:DescribeOptionGroups",
        "rds:DescribeOrderableDBInstanceOptions",
        "rds:DescribePendingMaintenanceActions",
        "rds:DescribeValidDBInstanceModifications",
        "rds:DownloadDBLogFilePortion",
        "rds:FailoverDBCluster",
        "rds:ListTagsForResource",
        "rds:ModifyDBCluster",
        "rds:ModifyDBClusterParameterGroup",
        "rds:ModifyDBClusterSnapshotAttribute",
        "rds:ModifyDBInstance",
        "rds:ModifyDBParameterGroup",
        "rds:ModifyDBSubnetGroup",
        "rds:ModifyEventSubscription",
        "rds:ModifyGlobalCluster",
        "rds:PromoteReadReplicaDBCluster",
        "rds:RebootDBInstance",
        "rds:RemoveFromGlobalCluster",
        "rds:RemoveRoleFromDBCluster",
        "rds:RemoveSourceIdentifierFromSubscription",
        "rds:RemoveTagsForResource",
        "rds:ResetDBClusterParameterGroup",
        "rds:ResetDBParameterGroup",
        "rds:RestoreDBClusterFromSnapshot",
        "rds:RestoreDBClusterToPointInTime"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "DependencySids",
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "cloudwatch:GetMetricData",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "ec2:AllocateAddress",
        "ec2:AssignIpv6Addresses",

```

```
"ec2:AssignPrivateIpAddresses",
"ec2:AssociateAddress",
"ec2:AssociateRouteTable",
"ec2:AssociateSubnetCidrBlock",
"ec2:AssociateVpcCidrBlock",
"ec2:AttachInternetGateway",
"ec2:AttachNetworkInterface",
"ec2:CreateCustomerGateway",
"ec2:CreateDefaultSubnet",
"ec2:CreateDefaultVpc",
"ec2:CreateInternetGateway",
"ec2:CreateNatGateway",
"ec2:CreateNetworkInterface",
"ec2:CreateRoute",
"ec2:CreateRouteTable",
"ec2:CreateSecurityGroup",
"ec2:CreateSubnet",
"ec2:CreateVpc",
"ec2:CreateVpcEndpoint",
"ec2:DescribeAccountAttributes",
"ec2:DescribeAddresses",
"ec2:DescribeAvailabilityZones",
"ec2:DescribeCustomerGateways",
"ec2:DescribeInstances",
"ec2:DescribeNatGateways",
"ec2:DescribeNetworkInterfaces",
"ec2:DescribePrefixLists",
"ec2:DescribeRouteTables",
"ec2:DescribeSecurityGroupReferences",
"ec2:DescribeSecurityGroups",
"ec2:DescribeSubnets",
"ec2:DescribeVpcAttribute",
"ec2:DescribeVpcEndpoints",
"ec2:DescribeVpcs",
"ec2:ModifyNetworkInterfaceAttribute",
"ec2:ModifySubnetAttribute",
"ec2:ModifyVpcAttribute",
"ec2:ModifyVpcEndpoint",
"kms:DescribeKey",
"kms:ListAliases",
"kms:ListKeyPolicies",
"kms:ListKeys",
"kms:ListRetirableGrants",
"logs:DescribeLogStreams",
```

```

        "logs:GetLogEvents",
        "sns:ListSubscriptions",
        "sns:ListTopics",
        "sns:Publish"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "DocdbSLRSid",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "rds.amazonaws.com"
        }
    }
},
{
    "Sid": "DocdbElasticSLRSid",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
        }
    }
}
]
}

```

AmazonDoc資料庫 ElasticReadOnlyAccess

此政策授予唯讀許可，讓使用者能夠在 Amazon DocumentDB 中檢視彈性叢集資訊。附加此政策的主體無法進行任何更新或刪除現有資源，也無法建立新的 Amazon DocumentDB 資源。例如，具有這些許可的主體可以檢視與其帳戶相關聯的叢集和組態清單，但無法變更任何叢集的組態或設定。此政策中的許可分組如下：

- Amazon DocumentDB 彈性叢集許可可讓您列出 Amazon DocumentDB 彈性叢集資源、描述資源，以及取得有關這些資源的資訊。
- CloudWatch 權限是用來驗證服務測量結果。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "docdb-elastic:ListClusters",
        "docdb-elastic:GetCluster",
        "docdb-elastic:ListClusterSnapshots",
        "docdb-elastic:GetClusterSnapshot",
        "docdb-elastic:ListTagsForResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
    }
  ]
}
```


AmazonDoc資料庫 ElasticFullAccess

此政策授予管理許可，允許主體完整存取 Amazon DocumentDB 彈性叢集的所有 Amazon DocumentDB 動作。

此原則會在條件範圍內使用 AWS 標籤 (<https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html>) 來限定資源的存取範圍。如果您使用的是密碼，則必須使用標籤鍵DocDBElasticFullAccess和標籤值來標記密碼。如果您使用的是客戶管理的金鑰，則必須使用標籤金鑰DocDBElasticFullAccess和標籤值加上標籤。


此政策中的許可分組如下：

- Amazon DocumentDB 彈性群集許可允許所有 Amazon DocumentDB 操作。
- 此政策中的某些 Amazon EC2 許可需要驗證 API 請求中傳遞的資源。這是為了確保 Amazon DocumentDB 能夠成功使用資源來佈建和維護叢集。此政策中的其餘 Amazon EC2 許可允許 Amazon DocumentDB 建立所需的 AWS 資源，讓您可以像虛擬私人雲端端點一樣連接到叢集。
- AWS KMS Amazon DocumentDB 需要許可才能使用傳遞的金鑰來加密和解密 Amazon DocumentDB 彈性叢集中的靜態資料。

 Note

客戶管理的金鑰必須具有包含金鑰DocDBElasticFullAccess和標籤值的標籤。

- SecretsManager 需要許可才能驗證指定的密碼，並使用它為 Amazon DocumentDB 彈性叢集設定管理員使用者。

 Note

使用的密碼必須有一個帶有密鑰DocDBElasticFullAccess和標籤值的標籤。

- 建立指標和記錄發佈所需的服務連結角色時，需要 IAM 許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DocdbElasticSid",
      "Effect": "Allow",
      "Action": [
        "docdb-elastic:CreateCluster",
        "docdb-elastic:UpdateCluster",
        "docdb-elastic:GetCluster",
        "docdb-elastic>DeleteCluster",
        "docdb-elastic:ListClusters",
        "docdb-elastic:CreateClusterSnapshot",
        "docdb-elastic:GetClusterSnapshot",
        "docdb-elastic>DeleteClusterSnapshot",
        "docdb-elastic:ListClusterSnapshots",
        "docdb-elastic:RestoreClusterFromSnapshot",
        "docdb-elastic:TagResource",
        "docdb-elastic:UntagResource",
        "docdb-elastic:ListTagsForResource",

```



```

        "docdb-elastic:CopyClusterSnapshot",
        "docdb-elastic:StartCluster",
        "docdb-elastic:StopCluster"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "EC2Sid",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateVpcEndpoint",
        "ec2:DescribeVpcEndpoints",
        "ec2>DeleteVpcEndpoints",
        "ec2:ModifyVpcEndpoint",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeAvailabilityZones",
        "secretsmanager:ListSecrets"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
        }
    }
},
{
    "Sid": "KMSSid",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "kms:ViaService": [

```

```

        "docdb-elastic.*.amazonaws.com"
    ],
    "aws:ResourceTag/DocDBElasticFullAccess": "*"
  }
},
{
  "Sid": "KMSGrantSid",
  "Effect": "Allow",
  "Action": [
    "kms:CreateGrant"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/DocDBElasticFullAccess": "*",
      "kms:ViaService": [
        "docdb-elastic.*.amazonaws.com"
      ]
    },
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
},
{
  "Sid": "SecretManagerSid",
  "Effect": "Allow",
  "Action": [
    "secretsmanager:ListSecretVersionIds",
    "secretsmanager:DescribeSecret",
    "secretsmanager:GetSecretValue",
    "secretsmanager:GetResourcePolicy"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "secretsmanager:ResourceTag/DocDBElasticFullAccess": "*"
    },
    "StringEquals": {
      "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
    }
  }
},

```

```

    {
      "Sid": "CloudwatchSid",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "SLRSid",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
        }
      }
    }
  ]
}

```

AmazonDoc資料庫 ElasticServiceRolePolicy

您無法附加AmazonDocDBElasticServiceRolePolicy至您的 AWS Identity and Access Management 實體。此政策附加至服務連結角色，可讓 Amazon DocumentDB 代表您執行動作。如需詳細資訊，請參閱 [彈性叢集中的服務連結角色](#)。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*",
      "Condition": {

```

```

    "StringEquals": {
      "cloudwatch:namespace": [
        "AWS/DocDB-Elastic"
      ]
    }
  }
}
]
}

```

Amazon DocumentDB 更新受 AWS 管政策

| 變更 | 描述 | 日期 |
|---|--|------------|
| AmazonDoc資料庫 ElasticFullAccess , AmazonDoc資料庫 ConsoleFullAccess -變更 | 已更新原則以新增啟動/停止叢集和複製叢集快照動作。 | 2/21/2024 |
| AmazonDoc資料庫 ElasticReadOnlyAccess , AmazonDoc資料庫 ElasticFullAccess -變更 | 已更新原則以新增cloudwatch:GetMetricData 動作。 | 6/21/2023 |
| AmazonDoc資料庫 ElasticReadOnlyAccess – 新政策 | 適用於 Amazon 文件資料庫彈性叢集的新受管政策 | 6/8/2023 |
| AmazonDoc資料庫 ElasticFullAccess – 新政策 | 適用於 Amazon 文件資料庫彈性叢集的新受管政策 | 6/5/2023 |
| AmazonDoc資料庫 ElasticServiceRolePolicy – 新政策 | Amazon DocumentDB 為 Amazon DocumentDB 彈性集群創建一個新的 AWS ServiceRoleForDoc DB 彈性服務鏈接角色 | 11/30/2022 |
| AmazonDoc資料庫 ConsoleFullAccess -變更 | 政策已更新以新增 Amazon DocumentDB 全域和彈性叢集許可 | 11/30/2022 |
| AmazonDoc資料庫 ConsoleFullAccess , AmazonDoc資料庫 | 服務啟動 | 1/19/2017 |

| 變更 | 描述 | 日期 |
|---|----|----|
| FullAccess, AmazonDoc資料庫 ReadOnlyAccess-新政策 | | |

Amazon DocumentDB API 許可：動作、資源和條件參考

當您設定對 [Amazon DocumentDB 使用基於身份的政策 \(IAM 政策\)](#) 並撰寫可附加至 IAM 身分 (身分型政策) 的許可政策時，請使用下列各節做為參考。

下面列出了每個 Amazon DocumentDB API 操作。清單中包括您可以授與執行動作之權限的對應動作、您可以授與權限的 AWS 資源，以及您可以包含的條件索引鍵以進行精細的存取控制。您可以在政策的 Action 欄位中指定動作、在政策的 Resource 欄位中指定資源值，以及在政策的 Condition 欄位中指定條件。如需條件的相關資訊，請參閱 [在政策中指定條件](#)。

您可以在 Amazon DocumentDB 政策中使用 AWS 寬條件金鑰來表示條件。如需完 AWS 整金鑰清單，請參閱 IAM 使用者指南中的可用 [金鑰](#)。

您可以使用 IAM 政策模擬器測試 IAM 政策。它會自動提供每個動作所需的資源和參數清單，包括 Amazon DocumentDB AWS 動作。IAM 政策模擬器會決定您指定的每個動作所需的許可。如需 IAM 政策模擬器的相關資訊，請參閱 [IAM 使用者指南中的使用 IAM 政策模擬器測試 IAM 政策](#)。

Note

若要指定動作，請使用後接 API 操作名稱的 rds: 字首 (例如，rds:CreateDBInstance)。

以下列出 Amazon RDS API 操作及其相關動作、資源和條件金鑰。


主題

- [Support 資源層級許可的 Amazon 文件資料庫動作](#)
- [不 Support 資源層級許可的亞馬遜文件資料庫動作](#)

Support 資源層級許可的 Amazon 文件資料庫動作

資源層級權限可讓您指定允許使用者執行動作的資源。Amazon DocumentDB 對資源級許可提供部分支持。這表示對於某些 Amazon DocumentDB 動作，您可以根據必須滿足的條件或允許使用者使用的特定資源，控制何時允許使用者使用這些動作。例如，您可以授予使用者只修改特定執行個體的許可。

以下列出 Amazon DocumentDB API 操作及其相關動作、資源和條件金鑰。

 Note

對於某些管理功能，Amazon DocumentDB 會使用與 Amazon RDS 共用的操作技術。

| Amazon DocumentDB API 操作和動作 | 資源 | 條件金鑰 |
|---|--|--------------------------|
| AddTagsToResource | 執行個體 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> | rds:db-tag |
| rds:AddTagsToResource | 子網路群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> | rds:subgrp-tag |
| ApplyPendingMaintenanceAction | 執行個體 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> | rds:db-tag |
| rds:ApplyPendingMaintenanceAction | | |
| 複製資料庫 ClusterSnapshot | 叢集快照 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i> | rds:cluster-snapshot-tag |
| rds:CopyDBClusterSnapshot | | |
| CreateDBCluster | 叢集 | rds:cluster-tag |
| rds>CreateDBCluster | | |

| Amazon DocumentDB API 操作和動作 | 資源 | 條件金鑰 |
|--|---|---|
| | arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-name</i> | |
| | 叢集參數群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i> | rds:cluster-pg-tag |
| | 子網路群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> | rds:subgrp-tag |
| 創建數據庫 ClusterParameterGroup rds:CreateDBClusterParameterGroup | 叢集參數群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i> | rds:cluster-pg-tag |
| 創建數據庫 ClusterSnapshot rds:CreateDBClusterSnapshot | 叢集 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-name</i> 叢集快照 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i> | rds:cluster-tag rds:cluster-snapshot-tag |

| Amazon DocumentDB API 操作和動作 | 資源 | 條件金鑰 |
|---|---|--------------------|
| CreateDBInstance | 執行個體 | rds:DatabaseClass |
| rds:CreateDBInstance | arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> | rds:db-tag |
| | 叢集 | rds:cluster-tag |
| | arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-name</i> | |
| 創建數據庫 SubnetGroup | 子網路群組 | rds:subgrp-tag |
| rds:CreateDBSubnetGroup | arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> | |
| DeleteDBInstance | 執行個體 | rds:db-tag |
| rds>DeleteDBInstance | arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> | |
| 刪除資料庫 SubnetGroup | 子網路群組 | rds:subgrp-tag |
| rds>DeleteDBSubnetGroup | arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> | |
| 描述 B ClusterParameterGroups | 叢集參數群組 | rds:cluster-pg-tag |
| rds:DescribeDBClusterParameterGroups | arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i> | |

| Amazon DocumentDB API 操作和動作 | 資源 | 條件金鑰 |
|--|---|---------------------------------|
| 描述 B ClusterParameters rds:DescribeDBClusterParameters | 叢集參數群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i> | rds:cluster-pg-tag |
| DescribeDBClusters rds:DescribeDBClusters | 叢集 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-instance-name</i> | rds:cluster-tag |
| 描述 B ClusterSnapshotAttributes rds:DescribeDBClusterSnapshotAttributes | 叢集快照 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i> | rds:cluster-snapshot-tag |
| 描述 B SubnetGroups rds:DescribeDBSubnetGroups | 子網路群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> | rds:subgrp-tag |
| DescribePendingMaintenanceActions rds:DescribePendingMaintenanceActions | 執行個體 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> | rds:DatabaseClass rds:db-tag |

| Amazon DocumentDB API 操作和動作 | 資源 | 條件金鑰 |
|--|---|--------------------|
| FailoverDBCluster rds:FailoverDBCluster | 叢集 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-instance-name</i> | rds:cluster-tag |
| ListTagsForResource rds:ListTagsForResource | 執行個體 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> | rds:db-tag |
| | 子網路群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> | rds:subgrp-tag |
| ModifyDBCluster rds:ModifyDBCluster | 叢集 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-name</i> | rds:cluster-tag |
| | 叢集參數群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i> | rds:cluster-pg-tag |
| 修改資料庫 ClusterParameterGroup rds:ModifyDBClusterParameterGroup | 叢集參數群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i> | rds:cluster-pg-tag |

| Amazon DocumentDB API 操作和動作 | 資源 | 條件金鑰 |
|--|--|---------------------------------|
| 修改資料庫 ClusterSnapshotAttribute rds:ModifyDBClusterSnapshotAttribute | 叢集快照 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i> | rds:cluster-snapshot-tag |
| ModifyDBInstance rds:ModifyDBInstance | 執行個體 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> | rds:DatabaseClass rds:db-tag |
| RebootDBInstance rds:RebootDBInstance | 執行個體 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> | rds:db-tag |
| RemoveTagsFromResource rds:RemoveTagsFromResource | 執行個體 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i> 子網路群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> | rds:db-tag rds:subgrp-tag |
| 重置資料庫 ClusterParameterGroup rds:ResetDBClusterParameterGroup | 叢集參數群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i> | rds:cluster-pg-tag |

| Amazon DocumentDB API 操作和動作 | 資源 | 條件金鑰 |
|--|--|--------------------------|
| 恢復 ClusterFromSnapshot rds:RestoreDBClusterFromSnapshot | 叢集 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-instance-name</i> | rds:cluster-tag |
| | 叢集快照 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i> | rds:cluster-snapshot-tag |
| 恢復 ClusterToPointInTime rds:RestoreDBClusterToPointInTime | 叢集 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-instance-name</i> | rds:cluster-tag |
| | 子網路群組 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i> | rds:subgrp-tag |

不 Support 資源層級許可的亞馬遜文件資料庫動作

您可以使用 IAM 政策中的所有 Amazon DocumentDB 動作，授予或拒絕使用者使用該動作的權限。不過，並非所有 Amazon DocumentDB 動作都支援資源層級許可，這可讓您指定可以執行動作的資源。下列 Amazon DocumentDB API 動作目前不支援資源層級許可。因此，若要在 IAM 政策中使用這些動作，您必須為陳述式中的元素使用 * 萬用字 Resource 元，授與使用者使用該動作所有資源的權限。

- rds:DescribeDBClusterSnapshots
- rds:DescribeDBInstances

Amazon DocumentDB 資料庫

在 Amazon DocumentDB 中，使用者搭配密碼對叢集進行驗證。每個叢集都有在叢集建立期間建立的主要登入認證。

Note

所有在 2020 年 3 月 26 日之前建立的新使用者均獲授予 `dbAdminAnyDatabase`、`readWriteAnyDatabase` 和 `clusterAdmin` 角色。建議您重新評估所有使用者並視需要修改角色，以對叢集中所有使用者強制執行最低權限。如需詳細資訊，請參閱[使用角色型存取控制存取資料庫](#)。

主要和 `serviceadmin` 使用者

新建立的 Amazon DocumentDB 叢集有兩個使用者：主要使用者和 `serviceadmin` 使用者。

主要使用者是具有權限的單一使用者，可以執行管理工作，並建立具有角色的其他使用者。當您第一次連線到 Amazon DocumentDB 叢集時，您必須使用主要登入資料進行驗證。建立 Amazon DocumentDB 叢集時，主要使用者會收到該叢集的這些管理許可，並被授與的角色 `root`。

建立叢集時，會隱含地建立 `serviceadmin` 使用者。每個 Amazon DocumentDB 叢集都有一個提供 AWS 管理叢集功能的 `serviceadmin` 使用者。您無法以 `serviceadmin` 身分進行登入、捨棄、重新命名、變更密碼或變更許可。任何這麼做的嘗試都會造成錯誤。

Note

無法刪除 Amazon DocumentDB 叢集的主要和 `serviceadmin` 使用者，且無法撤銷主要使用者的 `root` 角色。如果您忘記主要使用者密碼，可使用 AWS Management Console 或重設密碼 AWS CLI。

建立其他使用者

當您以主要使用者 (或具有該角色的任何使用者 `createUser`) 身分連線之後，您可以建立新使用者，如下所示。

```
db.createUser(
```

```
{
  user: "sample-user-1",
  pwd: "password123",
  roles:
    [{"db":"admin", "role":"dbAdminAnyDatabase" }]
}
```

若要檢視使用者詳細資訊，您可以使用 `show users` 命令，如下所示。您另外可以使用 `dropUser` 命令移除使用者。如需詳細資訊，請參閱 [一般命令](#)。

```
show users
{
  "_id" : "serviceadmin",
  "user" : "serviceadmin",
  "db" : "admin",
  "roles" : [
    {
      "role" : "root",
      "db" : "admin"
    }
  ]
},
{
  "_id" : "myPrimaryUser",
  "user" : "myPrimaryUser",
  "db" : "admin",
  "roles" : [
    {
      "role" : "root",
      "db" : "admin"
    }
  ]
},
{
  "_id" : "sample-user-1",
  "user" : "sample-user-1",
  "db" : "admin",
  "roles" : [
    {
```

```
        "role" : "dbAdminAnyDatabase",
        "db" : "admin"
    }
]
}
```

在此範例中，新使用者 `sample-user-1` 歸屬於 `admin` 資料庫。對於新用戶來說，總是如此。Amazon DocumentDB 沒有的概念，因此所有身份驗證都是在 `admin` 數據庫的上下文中執行的。 `authenticationDatabase`

建立使用者時，如果您在指定角色時省略該 `db` 欄位，Amazon DocumentDB 會隱含地將角色歸因於發出連線的資料庫。例如，如果您的連線是針對資料庫 `sample-database` 發出，並執行以下命令，則使用者 `sample-user-2` 將建立於 `admin` 資料庫中，並將具有資料庫 `sample-database` 的 `readWrite` 許可。

```
db.createUser(
  {
    user: "sample-user-2",
    pwd: "password123",
    roles:
      ["readWrite"]
  }
)
```

建立具有所有資料庫範圍的角色的使用者 (例如，`readInAnyDatabase`) 時，您必須在建立使用者時位於 `admin` 資料庫內容中，或是在建立使用者時明確地陳述角色的資料庫。

要切換資料庫的內容，您可以使用下面的命令。

```
use admin
```

若要進一步了解以角色為基礎的存取控制，以及在叢集中的使用者之間強制執行最低權限，請參閱 [使用角色型存取控制存取資料庫](#)。

自動輪替 Amazon DocumentDB 的密碼

您可以使用 AWS Secrets Manager，將程式碼中的硬式編碼登入資料 (包括密碼) 更換成機密管理員的 API 呼叫，以便透過程式設計方法來擷取機密。這有助於確保不讓某人研究您的程式碼而盜用秘密，因

為秘密根本不在程式碼中。此外，您可以設定 Secrets Manager，根據您指定的排程自動輪換秘密。這可讓您以短期秘密取代長期秘密，有助於大幅降低洩漏風險。

使用 Secrets Manager，您可以使用 Secrets Manager 提供的 AWS Lambda 功能，自動輪換 Amazon DocumentDB (即秘密)。

如需有關 Amazon DocumentDB (AWS Secrets Manager 和 Amazon DocumentDB 資料庫)，請參閱下列主題：

- [博客:如何旋轉 Amazon DocumentDB 和 Amazon Redshift 憑據AWS Secrets Manager](#)
- [什麼是AWS Secrets Manager？](#)
- [輪換 Amazon DocumentDB 的金鑰](#)

使用角色型存取控制存取資料庫

您可以使用 Amazon DocumentDB (與 MongoDB 相容性) 中的角色型存取控制 (RBAC)，限制使用者可以在資料庫上執行的動作的存取。RBAC 的運作方式是將一或多個角色授予使用者。這些角色決定了使用者可以對資料庫資源執行的作業。Amazon DocumentDB 目前支援在資料庫層級範圍設定的內建角色，例如、`read`、`readWrite` 和使用者定義的角色 `readAnyDatabaseClusterAdmin`，這些角色可以限定特定動作和精細資源，例如根據您的需求進行集合。

RBAC 的常見使用案例包括透過建立對叢集中資料庫或集合具有唯讀存取權限的使用者來強制執行最低權限，以及可讓單一使用者存取叢集中指定資料庫或集合的多租用戶應用程式設計。

Note

所有在 2020 年 3 月 26 日之前建立的新使用者均獲授予 `dbAdminAnyDatabase`、`readWriteAnyDatabase` 和 `clusterAdmin` 角色。建議您重新評估所有現有使用者，並視需要修改角色，以強制執行叢集的最低權限。

主題

- [RBAC 概念](#)
- [RBAC 內建角色入門](#)
- [開始使用 RBAC 使用者定義角色](#)
- [以使用者身分連接到 Amazon DocumentDB](#)
- [一般命令](#)

- [功能差異](#)
- [限制](#)
- [使用角色型存取控制存取資料庫](#)

RBAC 概念

以下是與以角色為基礎的存取控制相關的重要術語與概念。如需 Amazon 文件資料庫使用者的詳細資訊，請參閱。[Amazon DocumentDB 資料庫](#)

- 使用者 — 可向資料庫驗證並執行作業的個別實體。
- 密碼 — 用來驗證使用者的密碼。
- 角色 — 授權使用者對一或多個資料庫執行動作。
- 管理資料庫 — 儲存和授權使用者的資料庫。
- Database (**db**) — 叢集內的命名空間，其中包含用於儲存文件的集合。

以下命令會建立一個名為 `sample-user` 的使用者。

```
db.createUser({user: "sample-user", pwd: "abc123", roles: [{role: "read", db: "sample-database"}]})
```

在此範例中：

- `user: "sample-user"` — 指示使用者名稱。
- `pwd: "abc123"` — 指示使用者密碼。
- `role: "read", "db: "sample-database"` — 指示使用者 `sample-user` 在中具有讀取權限 `sample-database`。

```
db.createUser({user: "sample-user", pwd: "abc123", roles: [{role: "read", db: "sample-database"}]})
```

以下範例顯示在您透過 `db.getUser(sample-user)` 取得使用者 `sample-user` 後的輸出。在此範例中，使用者 `sample-user` 位於 `admin` 資料庫中，但具有 `sample-database` 資料庫的讀取角色。

```

{
  "_id" : "sample-user",
  "user" : "sample-user",
  "db" : "admin",
  "roles" : [
    {
      "db" : "sample-database",
      "role" : "read"
    }
  ]
}

```

← User ID

← Username

← All users created in the *admin* database

← User *sample-user* has read permissions in database *sample-database*

建立使用者時，如果您在指定角色時省略該db欄位，Amazon DocumentDB 會隱含地將角色歸因於發出連線的資料庫。例如，如果您的連線是針對資料庫 `sample-database` 發出，並執行以下命令，則使用者 `sample-user` 將建立於 `admin` 資料庫中，並將具有資料庫 `sample-database` 的 `readWrite` 許可。

```
db.createUser({user: "sample-user", pwd: "abc123", roles: ["readWrite"]})
```

此操作的輸出將會如下所示。

```

{
  "user":"sample-user",
  "roles":[
    {
      "db":"sample-database",
      "role":"readWrite"
    }
  ]
}

```

建立具有所有資料庫範圍的角色的使用者 (例如，`readAnyDatabase`) 時，您必須在建立使用者時位於 `admin` 資料庫內容中，或是在建立使用者時明確地陳述角色的資料庫。要對 `admin` 資料庫發出命令，您可以使用命令 `use admin`。如需詳細資訊，請參閱 [一般命令](#)。

RBAC 內建角色入門

為了協助您開始使用以角色為基礎的存取控制，本節會逐一解說透過為具有不同任務職能的三個使用者建立角色，強制最低權限的案例範例。

- `user1` 是一名新主管，需要能夠檢視和存取叢集中的所有資料庫。
- `user2` 是一名新員工，只需要存取同一個叢集中的一個資料庫 `sample-database-1`。
- `user3` 是現有員工，需要在同一個叢集中檢視和存取他們之前無法存取的不同資料庫 `sample-database-2`。

`user1` 和 `user2` 稍後離職，因此必須撤銷他們的存取權。

若要建立使用者並授予角色，您向叢集驗證的使用者必須具有可對 `createUser` 和 `grantRole` 執行動作的相關角色。例如，角色 `admin` 和 `userAdminAnyDatabase` 都可以授予這些能力。如需每個角色的動作，請參閱[使用角色型存取控制存取資料庫](#)。

Note

在 Amazon DocumentDB 中，無論您是否針對資料庫發出命令 `create` `get` `drop` `grant` `revoke`，所有使用者和角色操作 (例如、`create`、`drop`、`grant` 等) 都會在 `admin` 資料庫中隱含地執行。

首先，要了解叢集中目前有哪些使用者和角色，您可以執行 `show users` 命令，如下列範例所示。您會看到兩個使用者，`serviceadmin` 和叢集的主要使用者。這兩個使用者永遠存在且無法刪除。如需詳細資訊，請參閱[Amazon DocumentDB 資料庫](#)。

```
show users
```

對於 `user1`，使用下列命令建立對整個叢集中所有資料庫具有讀寫存取權的角色。

```
db.createUser({user: "user1", pwd: "abc123", roles: [{role: "readWriteAnyDatabase", db: "admin"}]})
```

此操作的輸出將會如下所示。

```
{
  "user": "user1",
  "roles": [
    {
      "role": "readWriteAnyDatabase",
      "db": "admin"
    }
  ]
}
```

```
]
}
```

對於 `user2`，使用下列命令建立對資料庫 `sample-database-1` 具有唯讀存取權的角色。

```
db.createUser({user: "user2", pwd: "abc123", roles: [{role: "read", db: "sample-
database-1"}]})
```

此操作的輸出將會如下所示。

```
{
  "user": "user2",
  "roles": [
    {
      "role": "read",
      "db": "sample-database-1"
    }
  ]
}
```

若要模擬 `user3` 為現有使用者的案例，請先建立使用者 `user3`，然後指派新角色給 `user3`。

```
db.createUser({user: "user3", pwd: "abc123", roles: [{role: "readWrite", db: "sample-
database-1"}]})
```

此操作的輸出將會如下所示。

```
{
  "user": "user3",
  "roles": [
    {
      "role": "readWrite",
      "db": "sample-database-1"
    }
  ]
}
```

建立使用者 `user3` 後，請將 `user3` 的角色 `read` 指派給 `sample-database-2`。

```
db.grantRolesToUser("user3", [{role: "read", db: "sample-database-2"}])
```

最後，`user1` 和 `user2` 都離職了，而需要撤銷他們對叢集的存取權。您可以透過捨棄使用者來做到這一點，如下所示。

```
db.dropUser("user1")
db.dropUser("user2")
```

若要確保所有使用者都具有適當的角色，您可以使用下列命令列出所有使用者。

```
show users
```

此操作的輸出將會如下所示。

```
{
  "_id": "serviceadmin",
  "user": "serviceadmin",
  "db": "admin",
  "roles": [
    {
      "db": "admin",
      "role": "root"
    }
  ]
}
{
  "_id": "master-user",
  "user": "master-user",
  "db": "admin",
  "roles": [
    {
      "db": "admin",
      "role": "root"
    }
  ]
}
{
  "_id": "user3",
  "user": "user3",
  "db": "admin",
  "roles": [
    {
      "db": "sample-database-2",
      "role": "read"
    }
  ]
}
```

```
    },
    {
      "db": "sample-database-1",
      "role": "readWrite"
    }
  ]
}
```

開始使用 RBAC 使用者定義角色

為了協助您開始使用使用者定義的角色，本節將引導您針對具有不同工作功能的三個使用者建立角色來強制執行最低權限的範例案例。

在此範例中，以下內容適用：

- user1 是一名新主管，需要能夠檢視和存取叢集中的所有資料庫。
- user2 是一名新員工，只需要在同一個集群中的一個資料庫中的「查找」操作。sample-database-1
- user3 是現有員工，需要查看和訪問特定集合，col2 在不同的資料庫中，他們sample-database-2之前沒有訪問權限，在同一個集群中。
- 對於 user1，使用下列命令建立對整個叢集中所有資料庫具有讀寫存取權的角色。

```
db.createUser(
  {
    user: "user1", pwd: "abc123",
    roles: [{role: "readWriteAnyDatabase", db: "admin"}]
  }
)
```

此操作的輸出將會如下所示。

```
{
  "user": "user1",
  "roles": [
    {
      "role": "readWriteAnyDatabase",
      "db": "admin"
    }
  ]
}
```

```
}
```

對於user2，使用下列命令建立具有資料庫sample-database-1中所有集合「find」權限的角色。請注意，此角色將確保任何關聯的使用者只能執行尋找查詢。

```
db.createRole(  
{  
  role: "findRole",  
  privileges: [  
    {  
      resource: {db: "sample-database-1", collection: ""}, actions: ["find"]  
    }  
  ],  
  roles: []  
}  
)
```

此操作的輸出將會如下所示。

```
{  
  "role": "findRole",  
  "privileges": [  
    {  
      "resource": {  
        "db": "sample-database-1",  
        "collection": ""  
      },  
      "actions": [  
        "find"  
      ]  
    }  
  ],  
  "roles": [  
  ]  
}
```

接下來，創建 user (user2) 並將最近創建的角色附加findRole到用戶。

```
db.createUser(  
{  
  user: "user2",  
  pwd: "abc123",
```

```
    roles: []
  })

  db.grantRolesToUser("user2",["findRole"])
```

要模擬現有用戶的user3場景，首先創建用戶user3，然後創建一個名為 CollectionRole 的新角色，我們將在接下來的步驟中使用它。user3

現在，您可以將新角色指定給user3。這個新角色將user3允許在sample-database-2中插入、更新、刪除和查找對特定集合 col2 的訪問權限。

```
db.createUser(
{
  user: "user3",
  pwd: "abc123",
  roles: []
})

db.createRole(
{
  role: "collectionRole",
  privileges: [
    {
      resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
"update", "insert", "remove"]
    },
  ],
  roles: []
}
)
```

此操作的輸出將會如下所示。

```
{
  "role":"collectionRole",
  "privileges":[
    {
      "resource":{"
        "db":"sample-database-2",
        "collection":"col2"
      },
      "actions":[
        "find",
```



```
        "update",
        "insert",
        "remove"
    ]
}
],
"roles":[

]
}
```

現在，用戶user3已經創建，您可以授予user3該角色collectionFind。

```
db.grantRolesToUser("user3",["collectionRole"])
```

最後，user1 和 user2 都離職了，而需要撤銷他們對叢集的存取權。您可以透過捨棄使用者來做到這一點，如下所示。

```
db.dropUser("user1")
db.dropUser("user2")
```

若要確保所有使用者都具有適當的角色，您可以使用下列命令列出所有使用者。

```
show users
```

此操作的輸出將會如下所示。

```
{
  "_id":"serviceadmin",
  "user":"serviceadmin",
  "db":"admin",
  "roles":[
    {
      "db":"admin",
      "role":"root"
    }
  ]
}
{
  "_id":"master-user",
  "user":"master-user",
  "db":"admin",
```

```
"roles":[
  {
    "db":"admin",
    "role":"root"
  }
]
}
{
  "_id":"user3",
  "user":"user3",
  "db":"admin",
  "roles":[
    {
      "db":"admin",
      "role":"collectionRole"
    }
  ]
}
```

以使用者身分連接到 Amazon DocumentDB

連線到 Amazon DocumentDB 叢集時，您可以在特定資料庫的環境中進行連線。根據預設，如果您未在連線字串中指定資料庫，則會自動連接到 test 資料庫內容中的叢集。會對 test 資料庫中的集合發出所有像 insert 和 find 之類的集合層級命令。

要查看您在上下文中的數據庫，或者（換句話說）發出命令，請使用 mongo shell 中的 db 命令，如下所示。

查詢：

```
db
```

輸出：

```
test
```

雖然預設連線可能在 test 資料庫的內容中，但這並不一定表示與連線關聯的使用者有權對 test 資料庫執行動作。在上述範例案例中，如果您以具有 sample-database-1 資料庫 readWrite 角色的使用者 user3 身分驗證，則連線的預設內容是 test 資料庫。不過，如果您嘗試將文件插入 test 資料庫上的集合中，您會收到授權失敗的錯誤訊息。這是因為該使用者未獲權在該資料庫上執行該命令，如下圖所示。

查詢：

```
db
```

輸出：

```
test
```

查詢：

```
db.col.insert({x:1})
```

輸出：

```
WriteCommandError({ "ok" : 0, "code" : 13, "errmsg" : "Authorization failure" })
```

如果您變更連接到 `sample-database-1` 資料庫的內容，則可以寫入使用者有權執行此操作的集合。

查詢：

```
use sample-database-1
```

輸出：

```
switched to db sample-database-1
```

查詢：

```
db.col.insert({x:1})
```

輸出：

```
WriteResult({ "nInserted" : 1})
```

當您對具有特定使用者的叢集進行驗證時，也可以在連線字串中指定資料庫。如此在使用者通過 `admin` 資料庫驗證之後就不必執行 `use` 命令。

以下連線字串會對 `admin` 資料庫驗證使用者，但連線的內容將是針對 `sample-database-1` 資料庫。

```
mongo "mongodb://user3:abc123@sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/sample-database-2"
```

一般命令

本節提供在 Amazon DocumentDB 中使用以角色為基礎的存取控制的常見命令範例。您必須位於 admin 資料庫的內容中，才能建立和修改使用者和角色。您可以使用 `use admin` 命令切換至 admin 資料庫。

Note

對使用者和角色的修改將隱含地發生在 admin 資料庫中。建立具有所有資料庫範圍之角色的使用者 (例如, `readAnyDatabase`) 時, 您必須在建立使用者時位於 admin 資料庫內容中 (亦即 `use admin`), 或是在建立使用者時明確地陳述角色的資料庫 (如本節中的範例 2 所示)。

範例 1：建立具有資料庫 read 角色的使用者 foo。

```
db.createUser({user: "readInFooBar", pwd: "abc123", roles: [{role: "read", db: "foo"}]})
```

此操作的輸出將會如下所示。

```
{
  "user": "readInFooBar",
  "roles": [
    {
      "role": "read",
      "db": "foo"
    }
  ]
}
```

範例 2：建立具有所有資料庫讀取權限的使用者。

```
db.createUser({user: "readAllDBs", pwd: "abc123", roles: [{role: "readAnyDatabase", db: "admin"}]})
```

此操作的輸出將會如下所示。

```
{
  "user": "readAllDBs",
  "roles": [
    {
      "role": "readAnyDatabase",
      "db": "admin"
    }
  ]
}
```

範例 3：將read角色授與新資料庫上的現有使用者。

```
db.grantRolesToUser("readInFooBar", [{role: "read", db: "bar"}])
```

範例 4：更新使用者的角色。

```
db.updateUser("readInFooBar", {roles: [{role: "read", db: "foo"}, {role: "read", db: "baz"}]})
```

範例 5：撤銷使用者對資料庫的存取權。

```
db.revokeRolesFromUser("readInFooBar", [{role: "read", db: "baz"}])
```

範例 6：描述內建角色。

```
db.getRole("read", {showPrivileges:true})
```

此操作的輸出將會如下所示。

```
{
  "role": "read",
  "db": "sample-database-1",
  "isBuiltin": true,
  "roles": [
  ],
  "inheritedRoles": [
  ],
  "privileges": [
    {
```

```
    "resource":{
      "db":"sample-database-1",
      "collection":""
    },
    "actions":[
      "changeStream",
      "collStats",
      "dbStats",
      "find",
      "killCursors",
      "listCollections",
      "listIndexes"
    ]
  }
],
"inheritedPrivileges":[
  {
    "resource":{
      "db":"sample-database-1",
      "collection":""
    },
    "actions":[
      "changeStream",
      "collStats",
      "dbStats",
      "find",
      "killCursors",
      "listCollections",
      "listIndexes"
    ]
  }
]
```

範例 7：從叢集中刪除使用者。

```
db.dropUser("readInFooBar")
```

此操作的輸出將會如下所示。

```
true
```

範例 8：建立具有特定集合之讀取和寫入權限的角色

```
db.createRole(  
{  
  role: "collectionRole",  
  privileges: [  
    {  
      resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",  
"update", "insert", "remove"]  
    }  
  ],  
  roles: []  
}  
)
```

此操作的輸出將會如下所示。

```
{  
  "role":"collectionRole",  
  "privileges":[  
    {  
      "resource":{  
        "db":"sample-database-2",  
        "collection":"col2"  
      },  
      "actions":[  
        "find",  
        "update",  
        "insert",  
        "remove"  
      ]  
    }  
  ],  
  "roles":[  
  ]  
}
```

範例 9：建立使用者並指派使用者定義的角色

```
db.createUser(  
{  
  user: "user3",  
  pwd: "abc123",  
  roles: []  
}
```

```
})  
  
db.grantRolesToUser("user3",["collectionRole"])
```

範例 10：將其他權限授與使用者定義的角色

```
db.grantPrivilegesToRole(  
  "collectionRole",  
  [  
    {  
      resource: { db: "sample-database-1", collection: "col1" },  
      actions: ["find", "update", "insert", "remove"]  
    }  
  ]  
)
```

範例 11：從使用者定義的角色中移除權限

```
db.revokePrivilegesFromRole(  
  "collectionRole",  
  [  
    {  
      resource: { db: "sample-database-1", collection: "col2" },  
      actions: ["find", "update", "insert", "remove"]  
    }  
  ]  
)
```

範例 12：更新現有使用者定義的角色

```
db.updateRole(  
  "collectionRole",  
  {  
    privileges: [  
      {  
        resource: {db: "sample-database-3", collection: "sample-collection-3"},  
        actions: ["find", "update", "insert", "remove"]  
      }  
    ],  
    roles: []  
  }  
)
```


功能差異

在 Amazon DocumentDB 中，使用者和角色定義會儲存在 admin 資料庫中，而使用者會根據資料庫進行驗證。admin 此功能不同於 MongoDB Community Edition，但與 MongoDB Atlas 一致。

Amazon DocumentDB 也支援變更串流，可提供叢集集合中發生的變更事件依時間順序順序排列。listChangeStreams 動作會在叢集層級 (也就是跨所有資料庫) 套用，而且 modifyChangeStreams 動作可在資料庫層級和叢集層級套用。

限制

下表包含 Amazon DocumentDB 中以角色為基礎的存取控制的限制。

| 描述 | 限制 |
|--------------|------|
| 每個叢集的使用者數目 | 1000 |
| 與使用者相關聯的角色數目 | 1000 |
| 使用者定義的角色數 | 100 |
| 與權限相關聯的資源數 | 100 |

使用角色型存取控制存取資料庫

您可以使用以角色為基礎的存取控制建立使用者並授予其一或多個角色，以決定使用者可以在資料庫或叢集中執行哪些操作。

以下是目前支援的內建角 Amazon DocumentDB。

Note

在 Amazon DocumentDB 4.0 和 5.0 中，ListCollection 和 ListDatabase 命令可以選擇性地使用 authorizedCollections 和 authorizedDatabases 參數列出使用者在需要和 listDatabase 角色的情況下有權存取的集合 listCollections 和資料庫。此外，使用者現在可以在不需要 KillCursor 角色的情況下殺死自己的游標。

Database user

| 角色名稱 | 描述 | 動作 |
|-----------|----------------------|---|
| read | 授予使用者對指定資料庫的讀取權限。 | changeStreams collStats dbStats find killCursors listIndexes listCollections |
| readWrite | 授予使用者對指定資料庫的讀取和寫入權限。 | read 許可中的所有動作。 createCollection dropCollection createIndex dropIndex insert killCursors listIndexes listCollections remove |

| 角色名稱 | 描述 | 動作 |
|------|----|--------|
| | | update |

Cluster user

| 角色名稱 | 描述 | 動作 |
|----------------------|------------------------------------|--|
| readAnyDatabase | 授予使用者對叢集中所有資料庫的讀取存取權。 | read 許可中的所有動作。 listChangeStreams listDatabases |
| readWriteAnyDatabase | 授予使用者對叢集中所有資料庫的讀寫存取權。 | readWrite 許可中的所有動作。 listChangeStreams listDatabases |
| userAdminAnyDatabase | 授予使用者指派和修改任何使用者對指定資料庫所具有的角色或許可的能力。 | changeCustomData changePassword createUser dropRole dropUser grantRole listDatabases revokeRole |

| 角色名稱 | 描述 | 動作 |
|--------------------|------------------------------|--|
| | | viewRole viewUser |
| dbAdminAnyDatabase | 授予使用者在任何指定的資料庫上執行資料庫管理角色的能力。 | dbAdmin 許可中的所有動作。 dropCollection listDatabases listChangeStreams modifyChangeStreams |

Superuser

| 角色名稱 | 描述 | 動作 |
|------|---|--|
| root | 授予使用者存取合併下列所有角色的資源與操作：readWriteAnyDatabase、dbAdminAnyDatabase、userAdminAnyDatabase、clusterAdmin、restore 和 backup。 | 來自 readWriteAnyDatabase、dbAdminAnyDatabase、userAdminAnyDatabase、clusterAdmin、restore 和 backup 的所有動作。 |

Database administrator

| 角色名稱 | 描述 | 動作 |
|---------|---|---|
| dbAdmin | 授予使用者在指定的資料庫上執行管理任務的能力。 | collMod collStats createCollection createIndex dropCollection dropDatabase dropIndex dbStats find killCursors listIndexes listCollections modifyChangeStreams |
| dbOwner | 透過結合角色 dbAdmin 和 readWrite ，授予使用者在指定的資料庫上執行任何管理任務的能力。 | 來自 dbAdmin 和 readWrite 的所有動作。 |

在 Amazon DocumentDB 中進行記錄和監控控制

Amazon DocumentDB 是否與 MongoDB 相容，可提供各種您可以監控的 CloudWatch 指標，可以判斷 Amazon DocumentDB 叢集和執行個體的健全。您可以使用各種工具來檢視 Amazon 文件資料庫指標，包括 Amazon DocumentDB 主控台、AWS CLI、亞馬遜 CloudWatch 主控台和 CloudWatch API。如需監控的詳細資訊，請參閱[Amazon DocumentDB](#)。

除了 Amazon CloudWatch 指標之外，您還可以使用效能分析工具來記錄在叢集上執行的操作的執行時間和詳細資訊。Profiler 適用於監控叢集上最慢的操作，以協助您改善個別查詢效能和整體叢集效能。啟用後，操作會記錄到 Amazon CloudWatch 日誌，您可以使用 CloudWatch 洞察分析、監控和存檔您的 Amazon DocumentDB 分析資料。如需詳細資訊，請參閱[分析亞馬遜文檔數據庫操作](#)。

Amazon DocumentDB 是與整合的一項服務 AWS CloudTrail，可提供由使用者、角色或 Amazon DocumentDB 中所採取之 MongoDB 作的記錄。AWS CloudTrail 將 Amazon DocumentDB 的所有 AWS CLI API 呼叫擷取為事件，包括來自 Amazon DocumentDB AWS Management Console Amazon DocumentDB 開發出的呼叫。如需詳細資訊，請參閱[使用記錄 Amazon DocumentDB API 呼叫 AWS CloudTrail](#)。

使用 Amazon DocumentDB，您可以稽核叢集中執行的事件。已記錄事件的範例包括成功和失敗的身分驗證嘗試、在資料庫中放入集合，或建立索引。根據預設，Amazon DocumentDB 上的稽核功能會停用，而且您必須選擇使用此功能。如需詳細資訊，請參閱[稽核 Amazon DocumentDB 事件](#)。

更新您的 Amazon DocumentDB TLS 證書

主題

- [更新您的應用程式和 Amazon DocumentDB 叢集](#)
- [故障診斷](#)
- [常見問答集](#)

亞馬遜文件資料庫叢集的憑證授權單位 (CA) 憑證將於 2024 年 8 月開始更新。如果您使用已啟用傳輸層安全性 (TLS) (預設設定) 的 Amazon DocumentDB 叢集，且尚未輪替用戶端應用程式和伺服器憑證，則需要執行下列步驟來緩解應用程式與 Amazon DocumentDB 叢集之間的連線問題。

- [步驟 1：下載新的 CA 憑證並更新您的應用程式](#)
- [步驟 2：更新伺服器憑證](#)

CA 和伺服器憑證已作為 Amazon DocumentDB 標準維護和安全最佳實務的一部分進行了更新。用戶端應用程式必須將新的 CA 憑證新增至其信任存放區，而且必須在此到期日之前更新現有 Amazon DocumentDB 執行個體，才能使用新的 CA 憑證。

更新您的應用程式和 Amazon DocumentDB 叢集

請遵循本節所述步驟更新您應用程式的憑證授權機構憑證套件 ([步驟 1](#)) 和您叢集的伺服器憑證 ([步驟 2](#))。將變更套用至您的生產環境前，我們強烈在開發或預備環境中測試這些步驟。

Note

您必須在每個擁有 Amazon DocumentDB 叢集的每個步驟 AWS 區域中完成步驟 1 和 2。

步驟 1：下載新的 CA 憑證並更新您的應用程式

下載新的 CA 憑證並更新您的應用程式，以使用新的 CA 憑證建立與 Amazon DocumentDB 的 TLS 連線。請從下列位置下載新的 CA 憑證套件：<https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem>。此操作會下載名為 `global-bundle.pem` 的檔案。

Note

如果您要存取同時包含舊 CA 憑證 (`rds-ca-2019-root.pem`) 和新 CA 憑證 (`rds-ca-rsa2048-g1`) 的金鑰儲存庫，請確認金鑰儲存庫是否選取。 `rds-ca-rsa2048-g1`、`rds-ca-rsa4096-g1` 和 `global-bundle`

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

接著更新您的應用程式，以使用新的憑證套件。新的 CA 服務包同時包含舊的 CA 憑證 (`rds-ca-2019`) 和新的 CA 憑證 (`rds-ca-rsa2048-g1`)。 `rds-ca-rsa` 只要在新的 CA 套件中有這兩個憑證，即可用兩個步驟更新您的應用程式和叢集。

如果要確認應用程式是否有在使用最新的憑證授權機構憑證套件，請參閱 [我如何確定自己使用的是最新的 CA 套件？](#)。如果您已經在應用程式中使用最新的憑證授權機構憑證套件，可以跳到步驟 2。

如需搭配您的應用程式使用 CA 套件的範例，請參閱 [對傳輸中資料進行加密](#) 和 [使用已啟用的 TLS 連線](#)。

Note

MongoDB Go Driver 1.2.1 目前只接受 `sslcertificateauthorityfile` 中的一個 CA 伺服器憑證。請參閱[使用已啟用的 TLS 連線](#)來了解如何在 TLS 啟用時，使用 Go 連線至 Amazon DocumentDB。

步驟 2：更新伺服器憑證

應用程式更新為使用新的 CA 服務包之後，下一個步驟是修改 Amazon DocumentDB 叢集中的每個執行個體來更新伺服器憑證。若要修改執行個體以使用新的伺服器憑證，請參閱下列指示。

Amazon DocumentDB 提供下列 CA 來簽署資料庫執行個體的資料庫伺服器憑證：

- `rds-ca-rsa2048-g1` — 在大多數地區使用具有 RSA 2048 私密金鑰演算法和 SHA256 簽署演算法的憑證授權單位。AWS 此 CA 支援自動伺服器憑證輪換。
- `rds-ca-rsa4096`—使用具有 RSA 4096 私密金鑰演算法和 SHA384 簽章演算法的憑證授權單位。此 CA 支援自動伺服器憑證輪換。

Note

若您使用 AWS CLI，能以 [describe-certificates](#) 檢視上方所列憑證授權機構的有效性。

這些 CA 憑證包含在區域和全域憑證套件中。當您將 `rds-ca-rsa 2048-g1` 或 `rds-ca-rsa 4096 g1` CA 與資料庫搭配使用時，Amazon DocumentDB 會管理資料庫上的資料庫伺服器憑證。Amazon DocumentDB 會在資料庫伺服器憑證到期前自動進行輪換（可能需要重新開機）。

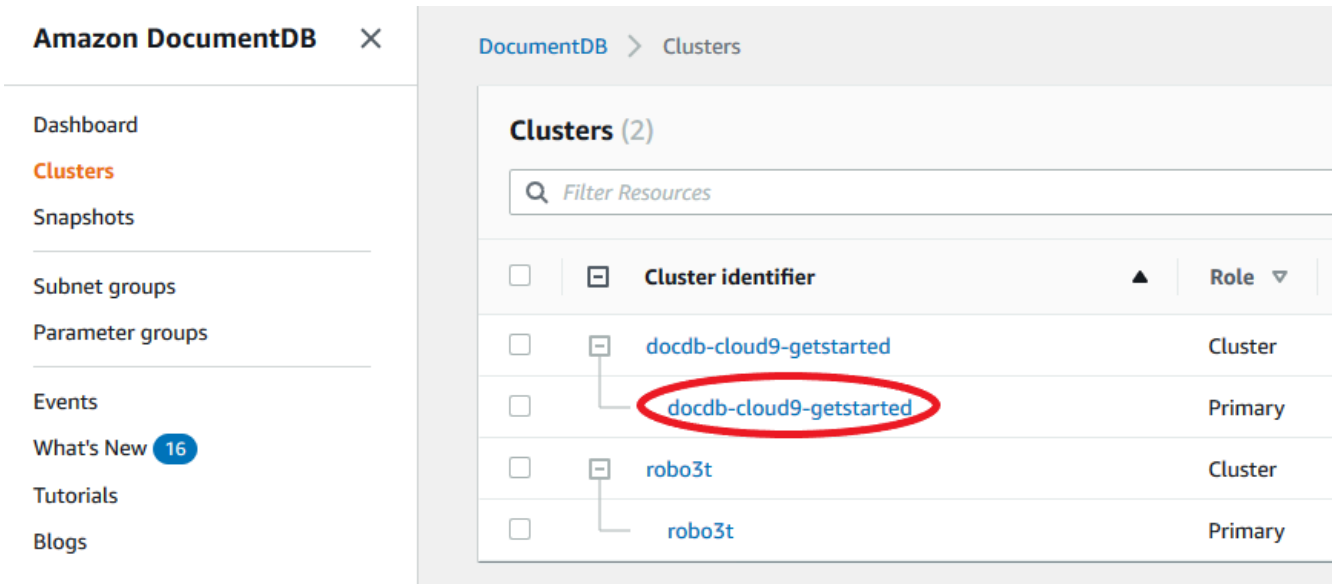
Note

更新您的執行個體需要重新開機，這可能會導致服務中斷。更新伺服器憑證前，請先確認您已完成[步驟 1](#)。

Using the AWS Management Console

完成下列步驟，以使用識別和輪換現有 Amazon DocumentDB 執行個體的舊伺服器憑證。AWS Management Console

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在畫面右上角的 [區域] 清單中，選擇叢集所AWS 區域在的位置。
3. 在主控台左側的導覽窗格中，選擇 [叢集]。
4. 您可能需要識別舊伺服器憑證上仍有哪些執行個體 (rds-ca-2019)。您可以在 [叢集] 表格最右側的 [憑證授權單位] 欄中執行此動作。
5. 在 [叢集] 資料表中，您會看到最左邊的 [叢集識別碼] 資料行。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。



6. 勾選您感興趣的執行個體左側的核取方塊。
7. 選擇 Actions (動作)，然後選擇 Modify (修改)。
8. 在 Certificate authority (憑證授權單位) 下方，選取此執行個體新的伺服器憑證 (rds-ca-rsa2048-g1)。
9. 您可以在下一頁上查看變更摘要。請注意，修改執行個體之前，會出現其他提醒來提醒您確保應用程式正使用最新的憑證 CA 套件，以免導致連線中斷。
10. 您可以在下次維護時段中套用修改，或立即套用。如果您想要立即修改伺服器憑證，請使用 Apple Immediately (立即套用) 選項。
11. 選擇 Modify instance (修改執行個體) 以完成更新。

Using the AWS CLI

完成下列步驟，以使用識別和輪換現有 Amazon DocumentDB 執行個體的舊伺服器憑證。AWS CLI

1. 若要立即修改執行個體，請對叢集中的每個執行個體執行下列命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
--ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. 若要修改叢集內的執行個體，以在叢集的下一個維護時段使用新的憑證授權機構憑證，請對叢集中的每個執行個體執行下列命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
--ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

故障診斷

如果您在憑證輪換的過程中無法連接到叢集，我們建議您執行下列動作：

- 重新啟動您的執行個體。輪換新憑證需要重新啟動每個執行個體。如果您將新憑證套用至一或多個執行個體，但未重新啟動它們，請重新啟動執行個體以套用新憑證。如需詳細資訊，請參閱[重新啟動 Amazon DocumentDB 實例](#)。
- 確認您的用戶端使用最新的憑證服務套件。請參閱 [我如何確定自己使用的是最新的 CA 套件？](#)。
- 確認您的執行個體使用最新的憑證。請參閱 [我如何知道我的 Amazon DocumentDB 執行個體正在使用舊/新伺服器憑證？](#)。
- 確認您的應用程式正在使用最新的憑證 CA。某些驅動程式 (例如 Java 和 Go) 需要額外的程式碼，才能將憑證套件中的多個憑證匯入信任存放區。如需使用 TLS 連線到亞馬遜文件資料庫的詳細資訊，請參閱 [以編程方式連接 Amazon DocumentDB](#)。
- 聯絡支援。如果您有任何疑問或問題，請聯繫 [AWS Support](#)。

常見問答集

以下是與 TLS 憑證相關的一些常見問題。

要是我有問題，該怎麼辦？

如果您有任何疑問或問題，請聯繫 [AWS Support](#)。

我如何知道我是否正在使用 TLS 連接到我的 Amazon DocumentDB 叢集？

您可以檢查叢集參數群組的 `tls` 參數，判斷您的叢集是否正在使用 TLS。如果 `tls` 參數設為 `enabled`，即表示您正在使用 TLS 憑證連線到叢集。如需詳細資訊，請參閱[管理 Amazon DocumentDB 叢集參數群組](#)。

為何需要更新 CA 和伺服器憑證？

Amazon DocumentDB CA 和伺服器憑證正在作為亞馬 Amazon DocumentDB 標準維護和安全最佳實務的一部分進行更新。目前的 CA 和伺服器憑證將於 2024 年 8 月開始到期。

如果我沒有在到期日前採取任何行動，會發生什麼情況？

如果您使用 TLS 連線到您的 Amazon DocumentDB 叢集，而且在則透過 TLS 連線的應用程式將無法再與 Amazon DocumentDB 叢集進行通訊。

Amazon DocumentDB 不會在到期前自動輪替您的資料庫憑證。您必須在到期日之前或之後更新應用程式和叢集，才能使用新的 CA 憑證。

我如何知道我的 Amazon DocumentDB 執行個體正在使用舊/新伺服器憑證？

若要識別仍使用舊伺服器憑證的 Amazon DocumentDB 執行個體，您可以使用 Amazon Document Management Console DB 或 AWS CLI

使用 AWS Management Console

辨識在您的叢集中仍在使用舊憑證的執行個體

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在畫面右上角的「區域」清單中，選擇執行個體所AWS 區域在的位置。
3. 在主控台左側的導覽窗格中，選擇 [叢集]。
4. 憑證授權單位欄 (靠近表格最右邊) 會顯示哪些執行處理仍在舊伺服器憑證 (`rds-ca-2019`) 和新伺服器憑證 (`rds-ca-rsa2048-g1`) 上。

使用 AWS CLI

若要辨識您的叢集中正使用舊伺服器憑證的執行個體，請搭配下列參數使用 `describe-db-clusters` 命令。

```
aws docdb describe-db-instances \  
  --filters Name=engine,Values=docdb \  
  --query 'DBInstances[*].  
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

如何修改 Amazon DocumentDB 叢集中的個別執行個體以更新伺服器憑證？

建議您同時為指定叢集中的所有執行個體更新伺服器憑證。若要修改叢集中的執行個體，您可以使用主控台或 AWS CLI。

Note

更新您的執行個體需要重新開機，這可能會導致服務中斷。更新伺服器憑證前，請先確認您已完成[步驟 1](#)。

使用 AWS Management Console

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在畫面右上角的 [區域] 清單中，選擇叢集所AWS 區域在的位置。
3. 在主控台左側的導覽窗格中，選擇 [叢集]。
4. 憑證授權單位欄 (靠近表格最右側) 會顯示哪些執行處理仍在舊伺服器憑證上 (rds-ca-2019)。
5. 在「叢集」表格的「叢集 ID」底下，選取要修改的執行處理。
6. 選擇 Actions (動作)，然後選擇 Modify (修改)。
7. 在 Certificate authority (憑證授權單位) 下方，選取此執行個體新的伺服器憑證 (rds-ca-rsa2048-g1)。
8. 您可以在下一頁上查看變更摘要。請注意，修改執行個體之前，會出現其他提醒來提醒您確保應用程式正使用最新的憑證 CA 套件，以免導致連線中斷。
9. 您可以在下次維護時段中套用修改，或立即套用。
10. 選擇 Modify instance (修改執行個體) 以完成更新。

使用 AWS CLI

完成下列步驟，以使用識別和輪換現有 Amazon DocumentDB 執行個體的舊伺服器憑證。AWS CLI

1. 若要立即修改執行個體，請對叢集中的每個執行個體執行下列命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. 若要修改叢集內的執行個體，以在叢集的下一個維護時段使用新的憑證授權機構憑證，請對叢集中的每個執行個體執行下列命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

如果我將新的執行個體新增至現有叢集，會發生什麼情況？

所有建立的新執行個體將會使用舊的伺服器憑證，且需要使用舊 CA 憑證的 TLS 連線。在 2024 年 1 月 25 日之後建立的任何新的 Amazon DocumentDB 執行個體都將預設使用新憑證 2048-g1。rds-ca-rsa

如果我的叢集發生執行個體替換或容錯移轉作業，會發生什麼情況？

如果您的叢集發生執行個體替換作業，建立的新執行個體會繼續使用執行個體之前使用的相同伺服器憑證。我們建議您同時為所有執行個體更新伺服器憑證。如果我的叢集發生容錯移轉作業，會在新的主要伺服器上會使用伺服器憑證。

如果我沒有使用 TLS 連線到我的叢集，我仍需更新各個執行個體嗎？

如果您沒有使用 TLS 連線到您的 Amazon DocumentDB 叢集，則不需要採取任何動作。

如果我沒有使用 TLS 連線到自己的叢集，但打算在未來這樣做，我該怎麼辦？

如果您在 2024 年 1 月之前建立叢集，請遵循上一節中的[步驟 1](#) 和[步驟 2](#)，以確保您的應用程式正在使用更新的 CA 服務包，而且每個 Amazon DocumentDB 執行個體都使用最新的伺服器憑證。如果您在 2024 年 1 月 25 日之後建立叢集，您的叢集將已經擁有最新的伺服器憑證 (rds-ca-rsa2048-g1)。若要驗證您的應用程式是否使用最新的 CA 套件，請參閱 [如果我沒有使用 TLS 連線到我的叢集，我仍需更新各個執行個體嗎？](#)

截止日期可以延長 2024 年 8 月以後嗎？

如果您的應用程式是透過 TLS 連線，則無法延長截止日期。

我如何確定自己使用的是最新的 CA 套件？

若要確認您擁有最新的套裝軟體，請使用下列指令。要運行這個命令，你必須有 Java 安裝和 Java 工具需要在你的殼的 PATH 變量。如需詳細資訊，請參閱[使用 Java](#)

macOS 系統和 Amazon Linux

```
keytool -printcert -v -file global-bundle.pem
```

Windows

```
keytool -printcert -v -file global-bundle.p7b
```

我為何在 CA 套件的名稱中看到「RDS」？

對於某些管理功能，例如憑證管理，Amazon DocumentDB 會使用與 Amazon Relational Database Service 服務 (Amazon RDS) 共用的操作技術。

新憑證何時過期？

新的伺服器憑證將過期 (一般而言)，如下所示：

- rds-ca-rsa2048- 二零二六一年到期
- rds-ca-rsa四零六-二一二一二十一年到期

如果我套用新的伺服器憑證，我可以還原為舊的伺服器憑證嗎？

如果您需要將執行個體還原為舊憑證，建議您對叢集中的所有執行個體都這樣做。您可以使用 AWS Management Console 或 AWS CLI 還原叢集中各執行個體的伺服器憑證。

使用 AWS Management Console

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在畫面右上角的 [區域] 清單中，選擇叢集所AWS 區域在的位置。
3. 在主控台左側的導覽窗格中，選擇 [叢集]。
4. 在「叢集」表格的「叢集 ID」底下，選取要修改的執行處理。選擇 Actions (動作)，然後選擇 Modify (修改)。
5. 在 Certificate authority (憑證授權單位) 下方，您可以選取舊的伺服器憑證 (rds-ca-2019)。

6. 選擇 Continue (繼續) 以檢視修改摘要。
7. 在此產生的頁面中，您可以選擇排程要在下一次維護時段中套用的修改項目，或立即套用修改項目。進行選取，並選擇 Modify instance (修改執行個體)。

Note

若您選擇立即套用您的修改項目，則系統會一併套用待定修改佇列內的任何變更。如果有任何待定修改需要停機才能執行，則選擇此選項可能會導致未預期的停機。

使用 AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-  
certificate-identifier rds-ca-2019 <--apply-immediately | --no-apply-immediately>
```

如果選擇 `--no-apply-immediately`，將在叢集的下一個維護時段套用變更。

如果我們從快照或時間點還原回復，是否會有新的伺服器憑證？

如果您在 2024 年 8 月之後 point-in-time 還原快照或執行還原，則建立的新叢集將使用新的 CA 憑證。

如果我無法從任何 Mac 作業系統直接連線到我的 Amazon DocumentDB 叢集，該怎麼辦？

Mac OS 已更新受信任憑證的需求。信任憑證現在的有效期必須在 397 天以內 (請參閱<https://support.apple.com/en-us/HT211025>)。

Note

在較新版本的 Mac OS 中觀察到此限制。

Amazon DocumentDB 執行個體憑證的有效期限超過四年，超過 Mac 作業系統上限。若要從執行 Mac OS 的電腦直接連線到 Amazon DocumentDB 叢集，您必須在建立 TLS 連線時允許無效的憑證。在此情況下，無效的憑證表示有效期超過 397 天。在連線到 Amazon DocumentDB 叢集時，您應該先瞭解這些風險，才允許無效的憑證。

若要使用從 Mac 作業系統連接到 Amazon DocumentDB 叢集 AWS CLI，請使用參數 `tlsAllowInvalidCertificates`。


```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017  
--tlsAllowInvalidCertificates
```

更新您的 Amazon DocumentDB TLS 證書- GovCloud (美國西部)

Note

此資訊僅適用於 GovCloud (美國西部) 地區的使用者。

亞馬遜文件資料庫 (具有 MongoDB 相容性) 叢集的憑證授權單位 (CA) 憑證將於 2022 年 5 月 18 日更新。如果您使用已啟用傳輸層安全性 (TLS) (預設設定) 的 Amazon DocumentDB 叢集，且尚未輪替用戶端應用程式和伺服器憑證，則需要執行下列步驟來緩解應用程式與 Amazon DocumentDB 叢集之間的連線問題。

- [步驟 1：下載新的 CA 憑證並更新您的應用程式](#)
- [步驟 2：更新伺服器憑證](#)

CA 和伺服器憑證已作為 Amazon DocumentDB 標準維護和安全最佳實務的一部分進行了更新。先前的 CA 憑證將於 2022 年 5 月 18 日到期。用戶端應用程式必須將新的 CA 憑證新增至其信任存放區，而且現有的 Amazon DocumentDB 執行個體必須在此到期日之前更新，才能使用新的 CA 憑證。

更新您的應用程式和 Amazon DocumentDB 叢集

請遵循本節所述步驟更新您應用程式的憑證授權機構憑證套件 ([步驟 1](#)) 和您叢集的伺服器憑證 ([步驟 2](#))。將變更套用至您的生產環境前，我們強烈在開發或預備環境中測試這些步驟。

Note

您必須在每個擁有 Amazon DocumentDB 叢集 AWS 區域的每個步驟中完成步驟 1 和 2。

步驟 1：下載新的 CA 憑證並更新您的應用程式

下載新的 CA 憑證並更新您的應用程式，以使用新的 CA 憑證建立與 Amazon DocumentDB 的 TLS 連線。請從下列位置下載新的 CA 憑證套件：<https://truststore.pki.us-gov-west-1.rds.amazonaws.com/>

[us-gov-west-1/us-gov-west-1-bundle.pem](https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem)。此操作會下載名為 `us-gov-west-1-bundle.pem` 的檔案。

Note

如果您正存取包含舊的 CA 憑證 (`rds-ca-2017-root.pem`) 和新的 CA 憑證 (`rds-ca-rsa4096-g1.pem`) 的金鑰存放區，請確認該金鑰存放區選取 `CA-RSA4096-G1`。

```
wget https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem
```

接著更新您的應用程式，以使用新的憑證套件。新的 CA 服務包包含舊的 CA 憑證和新的 CA 憑證 (`rds-ca-rsa4096-g1.pem`)。只要在新的 CA 套件中有這兩個憑證，即可用兩個步驟更新您的應用程式和叢集。

在 2021 年 12 月 21 日之後下載任何 CA 憑證組合包應使用新的 CA 憑證組合包。如果要確認應用程式是否有在使用最新的憑證授權機構憑證套件，請參閱[我如何確定自己使用的是最新的 CA 套件？](#)。如果您已經在應用程式中使用最新的憑證授權機構憑證套件，可以跳到步驟 2。

如需搭配您的應用程式使用 CA 套件的範例，請參閱[對傳輸中資料進行加密](#) 和 [使用已啟用的 TLS 連線](#)。

Note

MongoDB Go Driver 1.2.1 目前只接受 `sslcertificateauthorityfile` 中的一個 CA 伺服器憑證。請參閱[使用已啟用的 TLS 連線](#)來了解如何在 TLS 啟用時，使用 Go 連線至 Amazon DocumentDB。

步驟 2：更新伺服器憑證

應用程式更新為使用新的 CA 服務包之後，下一個步驟是修改 Amazon DocumentDB 叢集中的每個執行個體來更新伺服器憑證。若要修改執行個體以使用新的伺服器憑證，請參閱下列指示。

Note

更新您的執行個體需要重新開機，這可能會導致服務中斷。更新伺服器憑證前，請先確認您已完成[步驟 1](#)。

Using the AWS Management Console

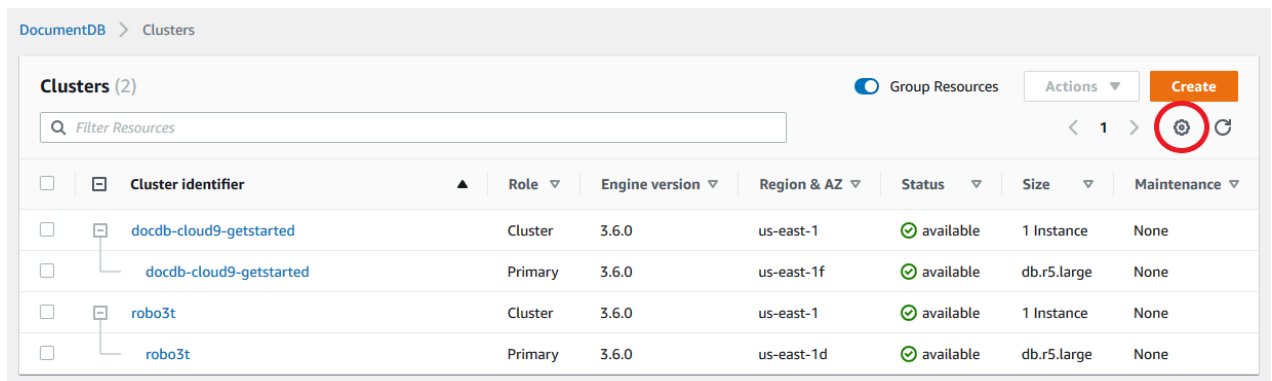
完成下列步驟，以使用識別和輪換現有 Amazon DocumentDB 執行個體的舊伺服器憑證AWS Management Console。

1. 登入AWS Management Console，開啟位於 <https://console.aws.amazon.com/docdb> 的主控台
主控台主控台主控台主控台 Amazon DocumentDB 控制台主控台主控台
2. 在畫面右上角的區域清單中，選擇您的叢集所在的叢集所在的叢集所AWS 區域在的叢集的所
在的叢集的所在
3. wh

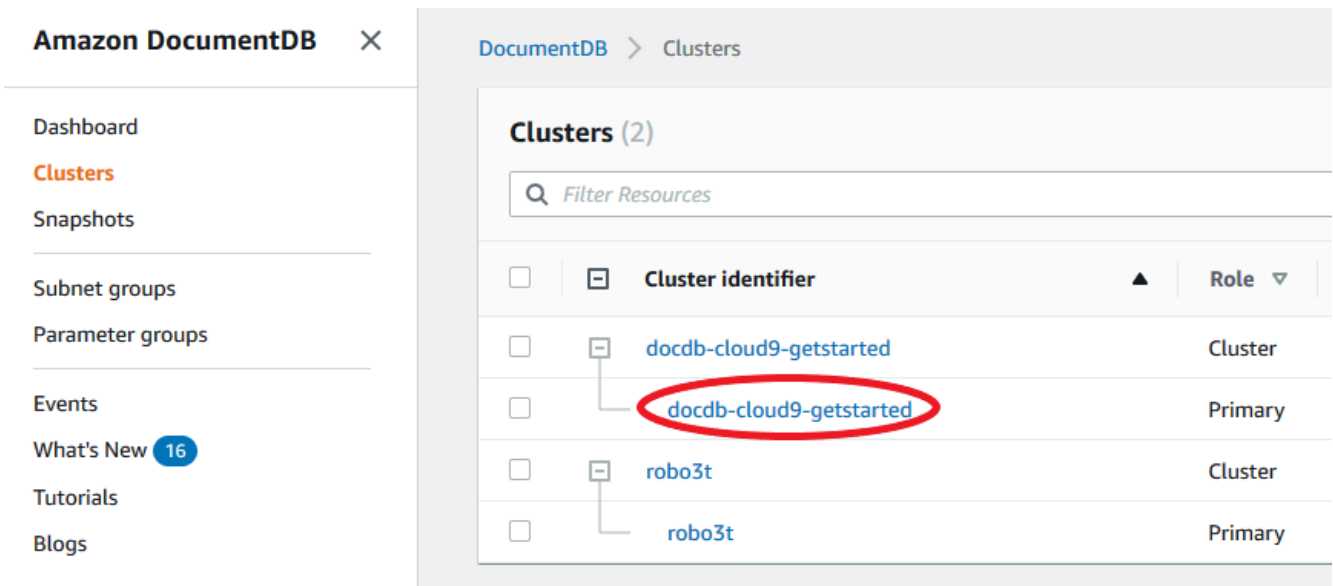
在主控台左側的導覽窗格中，選擇主控台左側的導覽窗格中的叢集。

4. 您可能需要識別舊伺服器憑證上仍有哪些執行個體 (rds-ca-2017)。您可以在預設為隱藏的憑證授權單位欄中執行此操作。若要顯示 Certificate authority (憑證授權單位) 資料欄，請執行下列操作：

- a. 選擇 Settings (設定) 圖示。



- b. 在可見資料欄清單下方，選擇 Certificate authority (憑證授權單位) 資料欄。
 - c. 選擇 Confirm (確認) 以儲存變更。
5. 現在回到 [叢集] 瀏覽方塊中，您會看到 [叢集識別碼] 資料行。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。



6. 核取位於您感興趣的執行個體左側的方塊。
7. 選擇 Actions (動作)，然後選擇 Modify (修改)。
8. 在 Certificate authority (憑證授權單位) 下方，選取此執行個體新的伺服器憑證 (rds-ca-rsa4096-g1)。
9. 您可以在下一頁上查看變更摘要。請注意，修改執行個體之前，會出現其他提醒來提醒您確保應用程式正使用最新的憑證 CA 套件，以免導致連線中斷。
10. 您可以在下次維護時段中套用修改，或立即套用。如果您想要立即修改伺服器憑證，請使用 Apple Immediately (立即套用) 選項。
11. 選擇 Modify instance (修改執行個體) 以完成更新。

Using the AWS CLI

完成下列步驟，以使用識別和輪換現有 Amazon DocumentDB 執行個體的舊伺服器憑證AWS CLI。

1. 若要立即修改執行個體，請對叢集中的每個執行個體執行下列命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
--ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. 若要修改叢集內的執行個體，以在叢集的下一個維護時段使用新的憑證授權機構憑證，請對叢集中的每個執行個體執行下列命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
--ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

疑難排解

如果您在憑證輪換的過程中無法連接到叢集，我們建議您執行下列動作：

- 重新啟動您的執行個體。輪換新憑證需要重新啟動每個執行個體。如果您將新憑證套用至一或多個執行個體，但未重新啟動它們，請重新啟動執行個體以套用新憑證。如需詳細資訊，請參閱[重新啟動 Amazon DocumentDB 實例](#)。
- 確認您的用戶端使用最新的憑證服務套件。請參閱[我如何確定自己使用的是最新的 CA 套件？](#)。
- 確認您的執行個體使用最新的憑證。請參閱[我如何知道我的 Amazon DocumentDB 執行個體正在使用舊/新伺服器憑證？](#)。
- 確認您的應用程式正在使用最新的憑證 CA。某些驅動程式 (例如 Java 和 Go) 需要額外的程式碼，才能將憑證套件中的多個憑證匯入信任存放區。如需使用 TLS 連接到 Amazon DocumentDB 的詳細資訊，請參閱[以編程方式連接 Amazon DocumentDB](#)。
- 聯絡支援。如果您有任何疑問或問題，請聯繫[AWS Support](#)。

常見問答集

以下是與 TLS 憑證相關的一些常見問題。

要是我有問題，該怎麼辦？

如果您有任何疑問或問題，請聯繫[AWS Support](#)。

我如何知道我是否正在使用 TLS 連接到我的 Amazon DocumentDB 叢集？

您可以檢查叢集參數群組的 `tls` 參數，判斷您的叢集是否正在使用 TLS。如果 `tls` 參數設為 `enabled`，即表示您正在使用 TLS 憑證連線到叢集。如需詳細資訊，請參閱[管理 Amazon DocumentDB 叢集參數群組](#)。

為何需要更新 CA 和伺服器憑證？

Amazon DocumentDB CA 和伺服器憑證已作為亞馬 Amazon DocumentDB 標準維護和安全最佳實務的一部分進行了更新。目前的 CA 和伺服器憑證將於 2022 年 5 月 18 日 (星期三) 到期。

如果不在到期日期之前採取任何行動在到期日期前的任何行情況

如果您使用 TLS 連線到您的 Amazon DocumentDB 叢集，而且在 2022 年 5 月 18 日之前未進行變更，則透過 TLS 連線的應用程式將無法再與 Amazon DocumentDB 叢集進行通訊。

Amazon DocumentDB 不會在到期前自動輪替您的資料庫憑證。您必須在到期日之前或之後更新應用程式和叢集，才能使用新的 CA 憑證。

我如何知道我的 Amazon DocumentDB 執行個體正在使用舊/新伺服器憑證？

若要識別仍使用舊伺服器憑證的 Amazon DocumentDB 執行個體，您可以使用 Amazon DocumentDBAWS Management Console 或AWS CLI。

使用 AWS Management Console

辨識在您的叢集中仍在使用舊憑證的執行個體

1. 登入AWS Management Console，開啟位於 <https://console.aws.amazon.com/docdb> 的主控台主控台主控台主控台 Amazon DocumentDB 控制台主控台主控台
2. 在畫面右上角的區域清單中，選擇您的執行個體駐留在的區域列表AWS 區域中，選擇要求的執行個體所在的
3. 在主控台左側的導覽窗格中，選擇 Instances (執行個體)。
4. Certificate authority (憑證授權單位) 資料欄 (預設為隱藏) 會顯示哪些執行個體仍在舊的伺服器憑證 (rds-ca-2017) 上，哪些仍在新的伺服器憑證上 (rds-ca-rsa4096-g1)。若要顯示 Certificate authority (憑證授權單位) 資料欄，請執行下列操作：
 - a. 選擇 Settings (設定) 圖示。
 - b. 在可見資料欄清單下方，選擇 Certificate authority (憑證授權單位) 資料欄。
 - c. 選擇 Confirm (確認) 以儲存變更。

使用 AWS CLI

若要辨識您的叢集中正使用舊伺服器憑證的執行個體，請搭配下列參數使用 describe-db-clusters 命令。

```
aws docdb describe-db-instances \  
  --filters Name=engine,Values=docdb \  
  --output text
```

```
--query 'DBInstances[*].  
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

如何修改 Amazon DocumentDB 叢集中的個別執行個體以更新伺服器憑證？

建議您同時為指定叢集中的所有執行個體更新伺服器憑證。若要修改叢集中的執行個體，您可以使用主控台或 AWS CLI。

Note

更新您的執行個體需要重新開機，這可能會導致服務中斷。更新伺服器憑證前，請先確認您已完成[步驟 1](#)。

使用 AWS Management Console

1. 登入AWS Management Console，開啟位於 <https://console.aws.amazon.com/docdb> 的主控台主控台主控台主控台 Amazon DocumentDB 控台主控台主控台
2. 在畫面右上角的區域清單中，選擇您的叢集所在的叢集所在的叢集所AWS 區域在的叢集的所在的叢集的所在
3. 在主控台左側的導覽窗格中，選擇 Instances (執行個體)。
4. Certificate authority (憑證授權機構) 欄 (預設為隱藏) 會顯示哪些執行個體仍在舊的伺服器憑證 (rds-ca-2017) 上。若要顯示 Certificate authority (憑證授權單位) 資料欄，請執行下列操作：
 - a. 選擇 Settings (設定) 圖示。
 - b. 在可見資料欄清單下方，選擇 Certificate authority (憑證授權單位) 資料欄。
 - c. 選擇 Confirm (確認) 以儲存變更。
5. 選取要修改的執行個體。
6. 選擇 Actions (動作)，然後選擇 Modify (修改)。
7. 在「憑證授權單位」下，選取此執行個體的新伺服器憑證 (rds-ca-rsa4096-g1)。
8. 您可以在下一頁上查看變更摘要。請注意，修改執行個體之前，會出現其他提醒來提醒您確保應用程式正使用最新的憑證 CA 套件，以免導致連線中斷。
9. 您可以在下次維護時段中套用修改，或立即套用。
10. 選擇 Modify instance (修改執行個體) 以完成更新。

使用 AWS CLI

完成下列步驟，以使用識別和輪換現有 Amazon DocumentDB 執行個體的舊伺服器憑證 AWS CLI。

1. 若要立即修改執行個體，請對叢集中的每個執行個體執行下列命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. 若要修改叢集內的執行個體，以在叢集的下一個維護時段使用新的憑證授權機構憑證，請對叢集中的每個執行個體執行下列命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

如果我將新的執行個體新增至現有叢集，會發生什麼情況？

所有建立的新執行個體將會使用舊的伺服器憑證，且需要使用舊 CA 憑證的 TLS 連線。任何在 2022 年 3 月 21 日之後建立的新 Amazon DocumentDB 執行個體都將預設為使用新憑證。

如果我的叢集發生執行個體替換或容錯移轉作業，會發生什麼情況？

如果您的叢集發生執行個體替換作業，建立的新執行個體會繼續使用執行個體之前使用的相同伺服器憑證。我們建議您同時為所有執行個體更新伺服器憑證。如果我的叢集發生容錯移轉作業，會在新的主要伺服器上會使用伺服器憑證。

如果我沒有使用 TLS 連線到我的叢集，我仍需更新各個執行個體嗎？

如果您沒有使用 TLS 連線到您的 Amazon DocumentDB 叢集，則不需要採取任何動作。

如果我沒有使用 TLS 連線到自己的叢集，但打算在未來這樣做，我該怎麼辦？

如果您在 2022 年 3 月 21 日之前建立叢集，請遵循上一節中的[步驟 1 和步驟 2](#)，以確保您的應用程式使用更新的 CA 服務包，而且每個 Amazon DocumentDB 執行個體都使用最新的伺服器憑證。如果您在 2022 年 3 月 21 日之後建立叢集，您的叢集將已經擁有最新的伺服器憑證。若要驗證您的應用程式是否使用最新的 CA 套件，請參閱[如果我沒有使用 TLS 連線到我的叢集，我仍需更新各個執行個體嗎？](#)

截止日期可以延長至 2022 年 5 月 18 日之後嗎？

如果您的申請是透過 TLS 連線，則截止日期無法延長至 2022 年 5 月 18 日之後。

我如何確定自己使用的是最新的 CA 套件？

由於相容性因素，舊的與新的 CA 套件檔案都會命名為 `us-gov-west-1-bundle.pem`。您也可以使用如 `openssl` 或 `keytool` 等工具來檢查 CA 套件。

我為何在 CA 套件的名稱中看到「RDS」？

對於某些管理功能 (例如，憑證管理) 的 Amazon DocumentDB 使用 Amazon RDS 共用的操作技術。

如果我套用新的伺服器憑證，我可以還原為舊的伺服器憑證嗎？

如果您需要將執行個體還原為舊憑證，建議您對叢集中的所有執行個體都這樣做。您可以使用 AWS Management Console 或 AWS CLI 還原叢集中各執行個體的伺服器憑證。

使用 AWS Management Console

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/docdb> 的主控台
2. 在畫面右上角的區域清單中，選擇您的叢集所在的 AWS 區域
3. 在主控台左側的導覽窗格中，選擇 Instances (執行個體)。
4. 選取要修改的執行個體。選擇 Actions (動作)，然後選擇 Modify (修改)。
5. 在「憑證授權單位」下，您可以選取舊的伺服器憑證 (`rds-ca-2017`)。
6. 選擇 Continue (繼續) 以檢視修改摘要。
7. 在此產生的頁面中，您可以選擇排程要在下一次維護時段中套用的修改項目，或立即套用修改項目。進行選取，並選擇 Modify instance (修改執行個體)。

Note

若您選擇立即套用您的修改項目，則系統會一併套用待修改佇列內的任何變更。如果有任何待修改需要停機才能執行，則選擇此選項可能會導致未預期的停機。

使用 AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-  
certificate-identifier rds-ca-2017 <--apply-immediately | --no-apply-immediately>
```


如果選擇 `--no-apply-immediately`，將在叢集的下一個維護時段套用變更。

如果我們從快照或時間點還原回復，是否會有新的伺服器憑證？

如果您在 2022 年 3 月 21 日之後 point-in-time 還原快照或執行還原，則建立的新叢集將使用新的 CA 憑證。

如果我有問題直接連接到我的 Amazon DocumentDB 叢集從 Mac OS X 卡塔利娜？

Mac OS X Catalina 已更新受信任憑證的需求。信任憑證現在的有效期限必須在 825 天以內 (請參閱<https://support.apple.com/en-us/HT210176>)。Amazon DocumentDB 執行個體憑證的有效期限為四年以上，超過 Mac OS X 的最大值。若要從執行 Mac OS X Catalina 的電腦直接連線至 Amazon DocumentDB 叢集，您必須在建立 TLS 連線時允許無效的憑證。在這種情況下，無效的憑證表示有效期間超過 825 天。在連線到 Amazon DocumentDB 叢集時，您應該先瞭解這些風險，然後才允許無效的憑證。

若要使用從 OS X 卡塔利娜連接到 Amazon DocumentDB 叢集 AWS CLI，請使用 `tlsAllowInvalidCertificates` 參數。

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017  
--tlsAllowInvalidCertificates
```

Amazon DocumentDB 中的合規驗證

由第三方稽核人員評估 Amazon DocumentDB 安全與合規性是多個 AWS 合規計劃中的一環，包括：

- 系統和組織控制 (SOC) 1、2、3。如需詳細資訊，請參閱 [SOC](#)。
- 支付卡產業資料安全標準 (PCI DSS)。如需詳細資訊，請參閱 [PCI DSS](#)。
- ISO 9001、27001、27017 和 27018。如需詳細資訊，請參閱 [ISO 認證](#)。
- 健康保險流通與責任法案商業夥伴協議 (HIPAA BAA)。如需詳細資訊，請參閱 [HIPAA 合規](#)。

AWS 會經常提供更新的 AWS 服務清單，旨在列出符合 [AWS 服務範圍合規計劃](#) 特定合規計劃範圍的服務。

您可以使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱在 [AWS Artifact 中下載報告](#)。

如需 AWS 合規計劃的詳細資訊，請參閱 [AWS 合規計劃](#)。

使用 Amazon DocumentDB 時的合規責任會取決於資料的敏感度、組織的合規目標，以及適用法律和法規。如果使用 Amazon DocumentDB 必須符合 HIPAA 或 PCI 等標準，會 AWS 提供資源予以協助：

- [AWS 合規資源](#)：這組手冊和指南可能適用於您的產業和位置。
- 《[安全與合規快速入門指南](#)》：這些部署指南討論在 AWS 上部署以安全及合規為重心之基準環境的架構考量和步驟。
- AWS 組 [Config](#)：此服務可評定資源組態與內部實務、業界準和法規的合規狀態。
- [AWS Security Hub](#)：AWS 中安全狀態的全面檢視，可助您檢查是否符合安全產業標準和最佳實務。
- [HIPAA 安全與合規架構白皮書 — 本白皮書](#)說明公司可如何運用 AWS 來建立 HIPAA 合規的應用程式。

Amazon DocumentDB 中的恢復能力

AWS 全球基礎架構是以 AWS 區域 與可用區域為中心建置的。AWS 區域 提供多個分開且隔離的實際可用區域，並以具備低延遲、高輸送量和高度備援特性的聯網相互連結。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

Amazon DocumentDB 叢集只能在至少兩個可用區域內有至少有兩個子網路的 Amazon VPC 中建立。Amazon DocumentDB 將叢集執行個體分散到至少兩個可用區域，有助於確保不惜可用區域出錯時您的叢集內有執行個體可用。Amazon DocumentDB 叢集的叢集磁盤區一律橫跨三個可用區域，以提供具有高耐用性的儲存，減少資料遺失的可能性。

如需 AWS 區域 與可用區域的詳細資訊，請參閱[AWS 全球基礎架構](#)。

除了 AWS Amazon DocumentDB 提供數種功能，可協助支援資料的彈性和備份需求。

容錯能力和自我修復儲存

儲存磁區的每個 10 GB 部分都有六種方法複製，跨三個可用區域。Amazon DocumentDB 使用的容錯儲存，可透明地處理最多兩個資料副本的遺失，不影響資料庫寫入可用性，以及最多三個副本，不影響讀取可用性。Amazon DocumentDB 儲存還具有自我修復功能；數據塊和磁盤會持續掃描錯誤並自動替換。

手動備份和還原

Amazon DocumentDB 提供可為叢集建立完整備份，以便長期保留和復原的功能。如需詳細資訊，請參閱 [在 Amazon DocumentDB 中備份和還原](#)。

時間點復原

時間點復原有助於保護您的 Amazon DocumentDB 叢集免遭意外寫入或刪除操作。有了時間點復原，就無需為建立、維護或排程隨需備份而煩惱。如需詳細資訊，請參閱[時間點還原](#)。

亞馬遜文檔數據庫的基礎設施安全

作為受管服務，亞馬遜文檔數據庫受到保護AWS全球網絡安全。如需有關 AWS 安全服務以及 AWS 如何保護基礎設施的詳細資訊，請參閱 [AWS 雲端安全](#)。若要使用基礎設施安全性的最佳實務來設計您的 AWS 環境，請參閱安全性支柱 AWS 架構良好的框架中的[基礎設施保護](#)。

您使用AWS透過網路存取亞馬遜文件資料庫的已發佈 API 呼叫。用戶端必須支援下列項目：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密 (PFS) 的密碼套件，例如 DHE (Ephemeral Diffie-Hellman) 或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取索引鍵 ID 和與 IAM 主體相關聯的私密存取索引鍵來簽署。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

您可以從任何網路位置呼叫這些 API 操作。您可以使用 Amazon 文件資料庫政策來控制來自特定 Amazon 虛擬私有雲端 (Amazon VPC) 端點或特定 VPC 的存取。實際上，這可以將對指定 Amazon DocumentDB 資源的網路存取從AWS網絡。

Note

亞馬遜文檔數據庫不支援以資源為基礎的存取政策。

Amazon DocumentDB 的安全最佳實務

為實現安全性最佳實務，您必須使用AWS Identity and Access Management(IAM) 帳戶控制對 Amazon DocumentDB API 操作的存取權限，特別是建立、修改或刪除 Amazon DocumentDB 資源的操作。這類資源包含叢集、安全群組和參數群組。您必須也使用 IAM 控制執行常見管理動作的動作，例如備份和還原叢集。建立 IAM 角色時，請採用最低權限的原則。

- 使用[角色類型存取控制](#)強制執行最低權限。

- 將個別 IAM 帳戶指派給管理 Amazon DocumentDB 資源的每個人。請勿使用 AWS 帳戶 root 用戶來管理 Amazon DocumentDB 資源。為每個人 (包含您) 建立 IAM 使用者。
- 授予每個 使用者執行其職責所需的最低許可集。
- 使用 IAM 群組來有效管理多個使用者的許可。如需 IAM 的詳細資訊，請參閱 [IAM 使用者指南](#)。如需 IAM 最佳實務的相關資訊，請參閱 [IAM 最佳實務](#)。
- 定期輪替您的 IAM 登入資料。
- 設定 AWS Secrets Manager 自動輪替 Amazon DocumentDB 的私密金鑰。如需詳細資訊，請參閱「[旋轉您的 AWS Secrets Manager 秘密](#)」和 [輪換 Amazon DocumentDB 的金鑰](#) 中的 AWS Secrets Manager 用戶指南。
- 使用 Transport Layer Security (TLS) 和靜態加密來加密您的資料。

稽核 Amazon DocumentDB 事件

使用 Amazon DocumentDB (與 MongoDB 相容性)，您可以稽核叢集中執行的事件。已記錄事件的範例包括成功和失敗的身分驗證嘗試、在資料庫中放入集合，或建立索引。根據預設，Amazon DocumentDB 上的稽核功能會停用，而且您必須選擇加入才能使用此功能。

啟用稽核功能後，Amazon DocumentDB 會將資料定義語言 (DDL)、資料操作語言 (DML)、身份驗證、授權和使用者管理事件記錄到 Amazon 日誌中。CloudWatch 啟用稽核功能後，Amazon DocumentDB 會將叢集的稽核記錄 (JSON 文件) 匯出至亞馬遜 CloudWatch 日誌。您可以使用亞馬遜 CloudWatch 日誌來分析、監控和存檔您的 Amazon DocumentDB 稽核事件。

雖然 Amazon DocumentDB 不會收取額外費用來啟用稽核功能，但是使用日誌時需支付標準費率。CloudWatch 如需 CloudWatch 日誌定價的相關資訊，請參閱 [Amazon CloudWatch 定價](#)。

Amazon DocumentDB 稽核功能與使用監控的服務資源使用量截然不同。AWS CloudTrail CloudTrail 記錄使用 AWS Command Line Interface (AWS CLI) 或 AWS Management Console 叢集、執行個體、參數群組和快照等資源執行的作業。依預設，使用的 AWS 資源稽核處 CloudTrail 於開啟狀態，且無法停用。Amazon DocumentDB 稽核功能是選擇加入的功能。它記錄您叢集內發生於物件的操作，例如資料庫、集合、索引和使用者。

主題

- [支援的事件](#)
- [啟用稽核](#)
- [停用稽核](#)

- [存取您的稽核活動](#)

支援的事件

Amazon DocumentDB 稽核支援下列事件類別：

- 數據定義語言 (DDL) -包括數據庫管理操作，連接，用戶管理和授權。
- 數據操縱語言讀取事件 (DML read) -包括find()和各種聚合運算符，算術運算符，布爾運算符和其他讀取查詢運算符。
- 數據操縱語言寫入事件 (DML 寫入) -包括insert(), update(), delete(),和bulkWrite()運算符

事件類型如下。

| 事件類型 | 類別 | 描述 |
|------------------|------------|-----------------------------------|
| authCheck | 授權 | 結果代碼 0：成功 結果代碼 13：未經授權的嘗試執行操作。 |
| authenticate | Connection | 成功或失敗的新連線身分驗證嘗試。 |
| createDatabase | DDL | 建立新的資料庫。 |
| createCollection | DDL | 在資料庫內建立新的集合。 |
| createIndex | DDL | 在集合內建立新的索引。 |
| dropCollection | DDL | 在資料庫內放入集合。 |
| dropDatabase | DDL | 放入資料庫。 |
| dropIndex | DDL | 在集合內放入索引。 |

| 事件類型 | 類別 | 描述 |
|--------------------------|----------------|----------------|
| modifyChangeStreams | DDL | 已建立變更串流。 |
| renameCollection | DDL | 重新命名資料庫中的集合。 |
| createRole | 角色管理 | 建立角色。 |
| dropAllRolesFromDatabase | 角色管理 | 刪除數據庫中的所有角色。 |
| dropRole | 角色管理 | 刪除角色。 |
| grantPrivilegesToRole | 角色管理 | 授與角色權限。 |
| grantRolesToRole | 角色管理 | 將角色授與使用者定義的角色。 |
| revokePrivilegesFromRole | 角色管理 | 撤銷角色的權限。 |
| revokeRolesFromRole | 角色管理 | 從使用者定義的角色撤銷角色。 |
| updateRole | 角色管理 | 更新角色。 |
| createUser | 使用者管理 | 建立新的使用者。 |
| dropAllUsersFromDatabase | 使用者管理 | 在資料庫內放入所有使用者。 |
| dropUser | 使用者管理 | 放入現有的使用者。 |
| grantRolesToUser | 使用者管理 | 將角色授與使用者。 |
| revokeRolesFromUser | 使用者管理 | 撤銷使用者的角色。 |
| updateUser | UserManagement | 更新現有的使用者。 |
| insert | DML 寫入 | 插入一個或多個文檔到集合中。 |

| 事件類型 | 類別 | 描述 |
|-------------------|----------------|------------------------------|
| delete | DML 寫入 | 從集合中刪除一個或多個文檔。 |
| update | DML 寫入 | 修改集合中的現有文檔或文檔。 |
| bulkWrite | DML 寫入 | 執行與執行順序控制多個寫操作。 |
| count | DML 讀取 | 傳回符合集合或檢視的 find () 查詢的文件計數。 |
| countDocuments | DML 讀取 | 傳回符合集合或檢視表查詢的文件計數。 |
| find | DML 讀取 | 選取集合或視圖中的文件，並將游標傳回到選取的文件。 |
| findAndModify | DML 讀取和 DML 寫入 | 修改並返回單個文檔。 |
| findOneAndDelete | DML 讀取和 DML 寫入 | 根據篩選條件和排序條件刪除單一文件，並傳回已刪除的文件。 |
| findOneAndReplace | DML 讀取和 DML 寫入 | 根據指定的篩選器取代單一文件。 |
| findOneAndUpdate | DML 讀取和 DML 寫入 | 根據篩選條件和排序準則更新單一文件。 |
| aggregate | DML 讀取和 DML 寫入 | 在彙總管線中支援 API。 |

| 事件類型 | 類別 | 描述 |
|----------|--------|--------------------------------|
| distinct | DML 讀取 | 在單個集合或視圖中查找指定字段的不同值，並在數組中返回結果。 |

Note

DML 事件文件參數欄位中的值有 1KB 的大小限制。Amazon DocumentDB 截斷，如果它超過 1 KB 的值。

Note

目前不會稽核 TTL 刪除事件。

啟用稽核

在叢集上啟用稽核的程序包含兩個步驟。請確定這兩個步驟都已完成，否則稽核記錄檔將不會傳送至 CloudWatch 記錄檔。

步驟 1. 啟用稽核記錄叢集參數


若要啟用稽核，您需要修改 `audit_logs` 參數群組中的參數。`audit_logs` 是要記錄的以逗號分隔的事件清單。事件必須以小寫指定，並且列表元素之間應該沒有空格。

您可以為參數群組設定下列值：

| 值 | 描述 |
|-----|--|
| ddl | 設定這項設定會啟用 DDL 事件的稽核功能，例如建立資料庫、刪除資料庫、建立集合、刪除索引、 |

| 值 | 描述 |
|-----------|--|
| | 刪除索引、AuthCheck、驗證、createUser、dropUser、使用者、使用者、updateUser 和 grantRolesTo revokeRolesFrom dropAllUsersFromDatabase |
| dml_read | 設置這將啟用 DML 讀取事件的審計，例如查找，排序計數，不同，組，項目，展開，地理交叉，地理內部和其他 MongoDB 讀取查詢運算符。 |
| dml_write | 設置這將啟用 DML 寫入事件的審計，例如插入 ()，更新 ()，刪除 () 和 bulkWrite 入 () |
| all | 設定此項將啟用資料庫事件的稽核功能，例如讀取查詢、寫入查詢、資料庫動作和管理員動作。 |
| none | 設定此選項將停用稽核 |

| 值 | 描述 |
|---------------|--|
| enabled (舊版) | 這是相當於 'ddl' 的舊式參數設定。設定這項設定會啟用 DDL 事件的稽核功能，例如建立資料庫、刪除資料庫、建立集合、刪除索引、刪除索引、AuthCheck、驗證、createUser、dropUser、使用者、使用者、updateUser 和。grantRolesTo revokeRolesFrom dropAllUsersFromDatabase我們不建議使用此設定，因為這是舊版設定。 |
| disabled (舊版) | 這是相當於「無」的舊式參數設定。我們不建議使用此設定，因為這是舊版設定。 |

 Note

稽核記錄叢集參數的預設值為 none (舊版 "disabled")。

您也可以組合使用上述值。

| 值 | 描述 |
|---------------------|--------------------------------|
| ddl, dml_read | 設定此項將啟用 DDL 事件和 DML 讀取事件的自動執行。 |
| ddl, dml_write | 設定此項將啟用 DDL 事件和 DML 寫入的自動化功能 |
| dml_read, dml_write | 設定此項將啟用所有 DML 事件的稽核功能 |

Note

您無法修改預設參數群組。

如需詳細資訊，請參閱下列內容：

- [建立 Amazon DocumentDB 叢集參數群組](#)

建立自訂參數群組之後，修改為將 `audit_logs` 參數值變更為 `enabled`。

- [修改 Amazon DocumentDB 叢集參數群組](#)

步驟 2. 啟用亞馬遜 CloudWatch 日誌匯出

當 `audit_logs` 叢集參數的值為 `enabled`、或時 `ddl` `dml_read`，您還必須啟用 Amazon DocumentDB `dml_write`，才能將日誌匯出到亞馬遜。CloudWatch 如果您省略其中一個步驟，則不會將稽核記錄傳送至 CloudWatch。

建立叢集、執行或還原快照時，您可以依照下列步驟啟用 CloudWatch 記錄。 `point-in-time-restore`

Using the AWS Management Console

若要啟用 Amazon DocumentDB 將日誌匯出到 CloudWatch 使用主控台，請參閱下列主題：

- 建立叢集時 — 在中[使用建立叢集和主要執行個體 AWS Management Console](#)，請參閱建立叢集：其他組態 (步驟 5, 記錄匯出)
- 修改現有叢集時 — [修改 Amazon DocumentDB 叢集](#)
- 執行叢集快照還原時 — [從叢集快照還原](#)
- 執行 point-in-time 還原時 — [時間點還原](#)

Using the AWS CLI

建立新叢集時啟用稽核日誌

下列程式碼會建立叢集 `sample-cluster` 並啟用 CloudWatch 稽核記錄。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --port 27017 \  
  --engine docdb \  
  --master-username master-username \  
  --master-user-password password \  
  --db-subnet-group-name default \  
  --enable-cloudwatch-logs-exports audit
```

針對 Windows：

```
aws docdb create-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --port 27017 ^  
  --engine docdb ^  
  --master-username master-username ^  
  --master-user-password password ^  
  --db-subnet-group-name default ^  
  --enable-cloudwatch-logs-exports audit
```

修改現有叢集時啟用稽核日誌

下列程式碼會修改叢集 `sample-cluster` 並啟用 CloudWatch 稽核記錄。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

針對 Windows：

```
aws docdb modify-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

這些操作的輸出類似以下內容 (JSON 格式)。

```
{  
  "DBCluster": {  
    "HostedZoneId": "ZNKXH85TT8WVW",  
    "StorageEncrypted": false,  
    "DBClusterParameterGroup": "default.docdb4.0",  
    "MasterUsername": "<user-name>",  
    "BackupRetentionPeriod": 1,  
    "Port": 27017,  
    "VpcSecurityGroups": [  
      {  
        "Status": "active",  
        "VpcSecurityGroupId": "sg-77186e0d"  
      }  
    ],  
    "DBClusterArn": "arn:aws:rds:us-east-1:900083794985:cluster:sample-cluster",  
    "Status": "creating",  
    "Engine": "docdb",  
    "EngineVersion": "4.0.0",  
    "MultiAZ": false,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1c",  
      "us-east-1f"  
    ],  
    "DBSubnetGroup": "default",  
    "DBClusterMembers": [],
```

```
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
    "EnabledCloudwatchLogsExports": [
      "audit"
    ],
    "PreferredMaintenanceWindow": "wed:03:08-wed:03:38",
    "AssociatedRoles": [],
    "ClusterCreateTime": "2019-02-13T16:35:04.756Z",
    "DbClusterResourceId": "cluster-Y0S52CUXGDTNKDQ7DH72I4LED4",
    "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
    "PreferredBackupWindow": "07:16-07:46",
    "DBClusterIdentifier": "sample-cluster"
  }
}
```

停用稽核

您可以停用 CloudWatch 記錄匯出並停用 `audit_logs` 參數來停用稽核。

停用 CloudWatch 記錄檔匯出

您可以使用 AWS Management Console 或 AWS CLI 停用匯出稽核日誌。

Using the AWS Management Console

下列程序會使用將日誌匯出 AWS Management Console 至停用 Amazon DocumentDB。
CloudWatch

停用稽核日誌

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇 Clusters (叢集)。接著對於您要停止匯出日誌的叢集，選擇其名稱左側的按鈕。
3. 選擇 Actions (動作)，然後選擇 Modify (修改)。
4. 向下捲動到 Log exports (日誌匯出) 區段，然後選擇 Disabled (停用)。
5. 選擇 Continue (繼續)。
6. 檢閱您的變更，然後選擇希望叢集套用此變更的時機。

- Apply during the next scheduled maintenance window (在下次排定的維護時段套用)
- Apply immediately (立即套用)

7. 選擇 Modify cluster (修改叢集)。

Using the AWS CLI

下列程式碼會修改叢集 `sample-cluster` 並停用 CloudWatch 稽核記錄。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

針對 Windows：

```
aws docdb modify-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBCluster": {  
    "DBClusterParameterGroup": "default.docdb4.0",  
    "HostedZoneId": "ZNKXH85TT8WVW",  
    "MasterUsername": "<user-name>",  
    "Status": "available",  
    "Engine": "docdb",  
    "Port": 27017,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1c",  
      "us-east-1f"  
    ],  
    "EarliestRestorableTime": "2019-02-13T16:35:50.387Z",  
    "DBSubnetGroup": "default",
```

```
    "LatestRestorableTime": "2019-02-13T16:35:50.387Z",
    "DBClusterArn": "arn:aws:rds:us-east-1:900083794985:cluster:sample-
cluster2",
    "Endpoint": "sample-cluster2.cluster-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
    "ReaderEndpoint": "sample-cluster2.cluster-ro-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
    "BackupRetentionPeriod": 1,
    "EngineVersion": "4.0.0",
    "MultiAZ": false,
    "ClusterCreateTime": "2019-02-13T16:35:04.756Z",
    "DBClusterIdentifier": "sample-cluster2",
    "AssociatedRoles": [],
    "PreferredBackupWindow": "07:16-07:46",
    "DbClusterResourceId": "cluster-Y0S52CUXGDTNKDQ7DH72I4LED4",
    "StorageEncrypted": false,
    "PreferredMaintenanceWindow": "wed:03:08-wed:03:38",
    "DBClusterMembers": [],
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ]
  }
}
```

停用 audit_logs 參數

若要停用叢集的 audit_logs 參數，您可以修改叢集，以便它使用 audit_logs 參數值為 disabled 的參數群組。或者，您可以修改叢集參數群組中的 audit_logs 參數值，使其成為 disabled。

如需詳細資訊，請參閱下列主題：

- [修改 Amazon DocumentDB 叢集](#)
- [修改 Amazon DocumentDB 叢集參數群組](#)

存取您的稽核活動

使用以下步驟存取 Amazon 上的稽核事件 CloudWatch。

1. 請在以下位置開啟 [CloudWatch 主控台](https://console.aws.amazon.com/cloudwatch/)。 <https://console.aws.amazon.com/cloudwatch/>
2. 請確定您所在的區域與 Amazon DocumentDB 叢集位於相同的區域。
3. 在導覽窗格中，選擇 Logs (日誌)。
4. 若要尋找叢集的稽核日誌，請從清單尋找並選擇 `/aws/docdb/yourClusterName/audit`。

您可在每個執行個體名稱下分別取得每個執行個體的稽核事件。

在 Amazon DocumentDB 中備份和還原

Amazon DocumentDB (與 MongoDB 相容性) 會持續將您的資料備份到亞馬遜簡單儲存服務 (Amazon S3) 1 到 35 天，以便您可以快速還原到備份保留期內的任何時間點。Amazon DocumentDB 也會自動擷取資料快照，做為此持續備份程序的一部分。

Note

這些是服務管理的 Amazon S3 儲存貯體，您將無法存取備份檔案。如果您想要控制自己的備份，請依照[傾印、還原、匯入和匯出資料](#)中的指示操作。

如果要將備份資料保留超過備份保留期，您也可以建立叢集資料的手動快照。備份過程不會影響您的叢集效能。

本節討論 Amazon DocumentDB 中備份功能的使用案例，並說明如何管理 Amazon DocumentDB 叢集的備份。

主題

- [備份與還原：概念](#)
- [了解備份儲存用量](#)
- [傾印、還原、匯入和匯出資料](#)
- [叢集快照考量](#)
- [比較自動與手動快照](#)
- [建立手動叢集快照](#)
- [複製 Amazon DocumentDB 叢集快照](#)
- [共用 Amazon DocumentDB 叢集快照](#)
- [從叢集快照還原](#)
- [時間點還原](#)
- [刪除叢集快照](#)

備份與還原：概念

| 名詞 | 描述 | API (動詞) |
|----------------------------------|--|---|
| Backup retention period (備份保留期間) | 您可以執行 point-in-time 還原的 1 到 35 天之間的一段時間。 | create-db-cluster modify-db-cluster restore-db-cluster-to-point-in-time |
| Amazon DocumentDB 存儲卷 | 高可用性和高耐用性的儲存磁碟區，可透過三個可用區域以六種方式複寫資料。無論叢集中有多少個執行個體，Amazon DocumentDB 叢集都具有高度耐用性。 | create-db-cluster delete-db-cluster |
| 備份時段 | 1 天內建立自動快照的期間。 | create-db-cluster describe-db-cluster modify-db-cluster |
| 自動快照 | 每日快照是叢集的完整備份，並由 Amazon | restore-db-cluster-from-snapshot describe-db-cluster-snapshot-attributes |

| 名詞 | 描述 | API (動詞) |
|------|----------------------------------|--|
| | DocumentDB 中的連續備份程序自動建立。 | <code>describe-db-cluster-snapshots</code> |
| 手動快照 | 您手動建立的快照，可在超過備份保留期之後用來保留叢集的完整備份。 | <code>create-db-cluster-snapshot</code> <code>copy-db-cluster-snapshot</code> <code>delete-db-cluster-snapshot</code> <code>describe-db-cluster-snapshot-attributes</code> <code>describe-db-cluster-snapshots</code> <code>modify-db-cluster-snapshot-attribute</code> |

了解備份儲存用量

Amazon DocumentDB 備份儲存包含備份保留期內的連續備份和保留期以外的手動快照。若要控制您的備份儲存用量，您可以降低備份保留間隔、移除不再需要的舊手動快照，或同時進行兩者。如需 Amazon 文件資料庫備份的一般資訊，請參閱 [在 Amazon DocumentDB 中備份和還原](#)。如需 Amazon DocumentDB 備份儲存的定價資訊，請參閱 [Amazon DocumentDB 定價](#)。

若要控制您的成本，您可以監控連續備份和超出保留期的手動快照所消耗的儲存量。然後，您可以降低備份保留間隔，並移除不再需要的手動快照。

您可以使用 Amazon CloudWatch 指標 `TotalBackupStorageBilledSnapshotStorageUsed`，以及檢閱 `BackupRetentionPeriodStorageUsed` 和監控 Amazon DocumentDB 備份所使用的儲存空間量，如下所示：

- `BackupRetentionPeriodStorageUsed` 表示目前用於儲存連續備份的備份儲存量。此指標值取決於叢集磁碟區的大小，以及您在保留期中所做的變更數。不過，為了方便計費，指標不會超過保留期內累積的叢集磁碟區大小。例如，如果您的叢集大小為 100 GiB，而保留期為兩天，則 `BackupRetentionPeriodStorageUsed` 的最大值為 200 GiB (100 GiB+100 GiB)。

- `SnapshotStorageUsed` 表示用於儲存在備份保留期外之手動快照的備份儲存量。在保留期內拍攝的手動快照不計入您的備份儲存。同樣地，自動快照不會計入您的備份儲存。每個快照的大小是您取得快照時的叢集磁碟區大小。`SnapshotStorageUsed` 值取決於您保留的快照數量和每個快照的大小。例如，假設您在保留期之外有一個快照，且叢集磁碟區在拍攝快照時的大小為 100 GiB。則 `SnapshotStorageUsed` 的數量為 100 GiB。
- `TotalBackupStorageBilled` 表示 `BackupRetentionPeriodStorageUsed` 和 `SnapshotStorageUsed` 的總和，減去相當於叢集磁碟區一天大小的免費備份儲存量。例如，如果您的叢集大小為 100 GiB，則您有一個保留天數，而且在保留期之外有一個快照，則 `TotalBackupStorageBilled` 為 100 GiB (100 GiB + 100 GiB - 100 GiB)。
- 這些指標是針對每個 Amazon DocumentDB 叢集獨立計算的。

您可以透過主控台監控 [Amazon DocumentDB 叢集](#) 並使用 [CloudWatch 指標](#) 建立報告。[CloudWatch](#) 如需如何使用 [CloudWatch](#) 量度的詳細資訊，請參閱 [Amazon DocumentDB](#)。

傾印、還原、匯入和匯出資料

您可以使用 `mongodump`、`mongorestore`、`mongoexport`、和 `mongoimport` 公用程式將資料移入和移出 Amazon DocumentDB 叢集。本節探討這些工具和組態的目的，以協助您達到更佳效能。

主題

- [mongodump](#)
- [mongorestore](#)
- [mongoexport](#)
- [mongoimport](#)
- [教學課程](#)

mongodump

該 `mongodump` 公用程式建立 MongoDB 資料庫的二進位 (BSON) 備份。由於以二進位格式儲存資料所達到的大小效率，因此此 `mongodump` 工具是在尋求將來源 MongoDB 部署還原到 Amazon DocumentDB 叢集時，將資料從來源 MongoDB 部署傾印出來的偏好方法。

視您用來執行命令的執行個體或機器上的可用資源而定，您mongodump可以使用此--numParallelCollections選項增加從預設 1 個印的 parallel 連線數目來加快速度。一個好的經驗法則是從 Amazon DocumentDB 叢集主要執行個體上的每個 vCPU 一個工作者開始。

Note

對於 Amazon DocumentDB，我們推薦 MongoDB 數據庫工具，並包括版本 100.6.1。您可以訪問 MongoDB 的數據庫工具下載[這裡](#)。

使用範例

以下是 Amazon DocumentDB 叢集中mongodump公用程式的使用範例。sample-cluster

```
mongodump --ssl \  
  --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \  
  --collection=sample-collection \  
  --db=sample-database \  
  --out=sample-output-file \  
  --numParallelCollections 4 \  
  --username=sample-user \  
  --password=abc0123 \  
  --sslCAFile global-bundle.pem
```

mongorestore

mongorestore 公用程式可讓您還原與 mongodump 公用程式一起建立的資料庫的二進位 (BSON) 備份。您可以在使用 --numInsertionWorkersPerCollection 選項還原期間，增加每個集合的工作者數目，提升還原效能 (預設為 1)。一個好的經驗法則是從 Amazon DocumentDB 叢集主要執行個體上的每個 vCPU 一個工作者開始。

使用範例

以下是 Amazon DocumentDB 叢集中mongorestore公用程式的使用範例。sample-cluster

```
mongorestore --ssl \  
  --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \  
  --username=sample-user \  
  --password=abc0123 \  
  --sslCAFile global-bundle.pem
```

```
--sslCAFile global-bundle.pem <fileToBeRestored>
```

mongoexport

該mongoexport工具將 Amazon DocumentDB 中的數據導出為 JSON , CSV 或 TSV 文件格式。mongoexport 工具是匯出需可人工或機器讀取之資料的偏好方法。

Note

mongoexport 不直接支援平行匯出。不過，您可以藉由同時針對不同的集合執行多個 mongoexport 任務來提高效能。

使用範例

以下是 Amazon DocumentDB 叢集中mongoexport工具的使用範例。sample-cluster

```
mongoexport --ssl \  
  --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \  
  --collection=sample-collection \  
  --db=sample-database \  
  --out=sample-output-file \  
  --username=sample-user \  
  --password=abc0123 \  
  --sslCAFile global-bundle.pem
```

mongoimport

此mongoimport工具會將 JSON、CSV 或 TSV 檔案的內容匯入 Amazon DocumentDB 叢集。您可以使用 --numInsertionWorkers 參數來平行化和加速匯入 (預設值為 1)。

使用範例

以下是 Amazon DocumentDB 叢集中mongoimport工具的使用範例。sample-cluster

```
mongoimport --ssl \  
  --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \  
  --collection=sample-collection \  
  --db=sample-database \  
  --file=<yourFile> \  
  --numInsertionWorkers=1
```

```
--numInsertionWorkers 4 \  
--username=sample-user \  
--password=abc0123 \  
--sslCAFile global-bundle.pem
```

教學課程

下列教學課程說明如何使用mongodump、mongoexport、mongoimport和mongoimport公用程式將資料移入和移出 Amazon DocumentDB 叢集。

1. 先決條件 — 在開始之前，請確保已佈建 Amazon DocumentDB 叢集，並且您可以存取與叢集相同 VPC 中的 Amazon EC2 執行個體。如需詳細資訊，請參閱 [使用 Amazon EC2 Connect](#)。

為了能夠使用 mongo 實用程序工具，您必須在 EC2 實例中安裝該 mongodb-org-tools 軟件包，如下所示。

```
sudo yum install mongodb-org-tools-4.0.18
```

由於 Amazon DocumentDB 預設使用傳輸層安全性 (TLS) 加密，因此您還必須下載 Amazon RDS 憑證授權單位 (CA) 檔案，以使用蒙戈殼層進行連線，如下所示。

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

2. 下載範例資料 — 在本教學課程中，您將下載一些包含餐廳相關資訊的範例資料。

```
wget https://raw.githubusercontent.com/ozlerhakan/mongodb-json-files/master/datasets/restaurant.json
```

3. 將範例資料匯入 Amazon 文件資料庫 — 由於資料採用邏輯 JSON 格式，因此您將使用公用mongoimport程式將資料匯入 Amazon DocumentDB 叢集。

```
mongoimport --ssl \  
  --host="tutorialCluster.amazonaws.com:27017" \  
  --collection=restaurants \  
  --db=business \  
  --file=restaurant.json \  
  --numInsertionWorkers 4 \  
  --username=<yourUsername> \  
  --password=<yourPassword> \  
  --sslCAFile global-bundle.pem
```


4. 使用傾印資料 **mongodump** — 現在 Amazon DocumentDB 叢集中有資料，您可以使用公用程式對該資料進行二進位轉儲。mongodump

```
mongodump --ssl \  
  --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017" \  
  --collection=restaurants \  
  --db=business \  
  --out=restaurantDump.bson \  
  --numParallelCollections 4 \  
  --username=<yourUsername> \  
  --password=<yourPassword> \  
  --sslCAFile global-bundle.pem
```

5. 刪除 **restaurants** 集合 — 在您還原 **business** 資料庫中的 **restaurants** 集合之前，您必須先卸除該資料庫中已存在的集合，如下所示。

```
use business
```

```
db.restaurants.drop()
```

6. 使用還原資料 **mongorestore** — 使用步驟 3 中資料的二進位轉儲，您現在可以使用該 **mongorestore** 公用程式將資料還原到 Amazon DocumentDB 叢集。

```
mongorestore --ssl \  
  --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017" \  
  --numParallelCollections 4 \  
  --username=<yourUsername> \  
  --password=<yourPassword> \  
  --sslCAFile global-bundle.pem restaurantDump.bson
```

7. 匯出資料使用 **mongoexport** — 若要完成教學課程，請以 JSON 檔案的格式從叢集匯出資料，與您在步驟 1 中匯入的檔案沒有什麼不同。

```
mongoexport --ssl \  
  --host="tutorialCluster.node.us-east-1.docdb.amazonaws.com:27017" \  
  --collection=restaurants \  
  --db=business \  
  --out=restaurant2.json \  
  --username=<yourUsername> \  
  --password=<yourPassword> \  
  --sslCAFile global-bundle.pem
```

8. 驗證 — 您可以使用下列命令驗證步驟 5 的輸出產生與步驟 1 相同的結果。

```
wc -l restaurant.json
```

從這個命令的輸出：

```
2548 restaurant.json
```

```
wc -l restaurant2.json
```

從這個命令的輸出：

```
2548 restaurant2.json
```

叢集快照考量

Amazon DocumentDB 會在叢集的備份時段期間，為您的叢集建立每日自動快照。Amazon DocumentDB 會根據您指定的備份保留期，儲存叢集的自動快照。如有需要，您可將叢集復原至備份保留期間的任何時間點。在相同叢集的不同區域執行複製操作時，自動快照不會發生。

主題

- [備份儲存](#)
- [備份時段](#)
- [備份保留期間](#)
- [複製叢集快照加密](#)

除了自動叢集快照，您也可以手動建立叢集快照。您可以複製自動與手動快照。如需詳細資訊，請參閱 [建立手動叢集快照](#) 及 [複製 Amazon DocumentDB 叢集快照](#)。

Note

您的叢集必須為可用狀態，才能建立自動快照。

您無法共用 Amazon DocumentDB 自動化叢集快照。因應措施是，您可以藉由複製自動快照來建立手動快照，然後共用該複本。如需複製快照的詳細資訊，請參閱 [複製 Amazon DocumentDB 叢集快照](#)。如需從快照還原叢集的詳細資訊，請參閱 [從叢集快照還原](#)。

備份儲存

每個備份儲存的 Amazon DocumentDB 備份儲存都 AWS 區域 是由備份保留期所需的備份儲存組成，包括該區域中的自動和手動叢集快照。預設備份保留期間為 1 天。如需備份儲存定價的詳細資訊，請參閱 [Amazon DocumentDB 定價](#)。

刪除叢集時，所有自動快照將一併刪除，且無法復原。但是，刪除叢集時不會刪除手動快照。如果您選擇讓 Amazon DocumentDB 在刪除叢集之前建立最終快照 (手動快照)，則可以使用最終快照來復原叢集。

如需快照與儲存的詳細資訊，請參閱 [了解備份儲存用量](#)。

備份時段

自動快照會每天在您偏好的備份時段內執行。如果快照需要的時間超過所分配的備份時段，備份會在時段結束後繼續執行直到完成。備份時段不可與每週叢集維護時段重疊。

如果您在建立叢集時未指定偏好的備份時段，Amazon DocumentDB 會指派預設的 30 分鐘備份時段。這個時段是根據與叢集區域相關聯的 8 小時時段內隨機選擇。您可以修改叢集來變更偏好的備份時段。如需詳細資訊，請參閱 [修改 Amazon DocumentDB 叢集](#)。

| 區域名稱 | 區域 | UTC 時間區塊 |
|---------------|----------------|-----------------|
| 美國東部 (俄亥俄) | us-east-2 | 03:00-11:00 |
| 美國東部 (維吉尼亞北部) | us-east-1 | 03:00-11:00 |
| 美國西部 (奧勒岡) | us-west-2 | 06:00-14:00 |
| 亞太區域 (香港) | ap-east-1 | 06:00-14:00 |
| 亞太區域 (海德拉巴) | ap-south-2 | 上午 6 時半至下午 2 時半 |
| 亞太區域 (孟買) | ap-south-1 | 06:00-14:00 |
| 亞太區域 (首爾) | ap-northeast-2 | 13:00-21:00 |
| 亞太區域 (新加坡) | ap-southeast-1 | 下午 10 時 |
| 亞太區域 (悉尼) | ap-southeast-2 | 下午八時 |

| 區域名稱 | 區域 | UTC 時間區塊 |
|---------------------|----------------|---------------|
| 亞太區域 (東京) | ap-northeast-1 | 13:00-21:00 |
| 加拿大 (中部) | ca-central-1 | 03:00-11:00 |
| 中國 (北京) | cn-north-1 | 06:00-14:00 |
| 中國 (寧夏) | cn-northwest-1 | 06:00-14:00 |
| 歐洲 (法蘭克福) | eu-central-1 | 下午 9 時至 5 時 |
| 歐洲 (愛爾蘭) | eu-west-1 | 22:00-06:00 |
| 歐洲 (倫敦) | eu-west-2 | 22:00-06:00 |
| 歐洲 (米蘭) | eu-south-1 | 02:00-10 點 |
| 歐洲 (巴黎) | eu-west-3 | 23:59-07:29 |
| 中東 (阿拉伯聯合大公國) | me-central-1 | 下午 5 時至下午 1 時 |
| 南美洲 (聖保羅) | sa-east-1 | 凌晨八時 |
| AWS GovCloud (美國東部) | us-gov-east-1 | 下午一時 |
| AWS GovCloud (美國西部) | us-gov-west-1 | 06:00-14:00 |

備份保留期間

備份保留期間是指自動備份在自動刪除之前保留的天數。Amazon DocumentDB 支援 1-35 天的備份保留期。

您可以在建立叢集時設定備份保留期。如果您沒有明確設定備份保留期，則會為叢集指派預設的備份保留期 1 天。建立叢集之後，您可以使用 AWS Management Console 或修改叢集來修改備份保留期間 AWS CLI。如需詳細資訊，請參閱 [修改 Amazon DocumentDB 叢集](#)。

複製叢集快照加密

叢集和快照加密是以 KMS 加密金鑰為基礎。KMS 金鑰 ID 是 KMS 加密金鑰的 Amazon Resource Name (ARN)、KMS 金鑰識別符或 KMS 金鑰別名。

下列準則與限制適用：

- 建立快照時，會從叢集推斷加密。如果叢集已加密，則會使用相同的 KMS 金鑰加密該叢集的快照。如果叢集未加密，則不會加密快照。
- 如果您從 Amazon Web Services 帳戶複製加密的叢集快照，您可以指定一個值，以 KmsKeyId 便使用新的 KMS 加密金鑰加密複本。如果未指定的值 KmsKeyId，則叢集快照的副本會使用與來源叢集快照相同的 KMS 金鑰加密。
- 如果您複製從另一個 Amazon Web Services 帳戶共用的加密叢集快照，則必須指定的值 KmsKeyId。
- 若要將加密的叢集快照複製到另一個 Amazon Web Services 區域，請設定 KmsKeyId 為您要用來加密目的地區域中叢集快照副本的 KMS 金鑰 ID。KMS 加密金鑰是建立於其所在 Amazon Web Services 區域的特定金鑰，而且您無法使用來自另一個 Amazon Web Services 區域中某個 Amazon Web Services 區域的加密金鑰。
- 如果您複製未加密的叢集快照並指定 KmsKeyId 參數值，則會傳回錯誤。

比較自動與手動快照

以下是 Amazon DocumentDB (與 MongoDB 兼容性) 的自動和手動快照的主要功能。

Amazon DocumentDB 自動快照具有下列主要功能：

- 自動快照命名 — 自動快照名稱遵循模式 `rds:<cluster-name>-yyyy-mm-dd-hh-mm`，`yyyy-mm-dd-hh-mm` 代表建立快照的日期和時間。
- 依排程自動建立 — 建立或修改叢集時，您可以將備份保留期設定為 1 到 35 天的整數值。根據預設，新叢集的備份保留期為 1 天。備份保留期定義自動刪除自動快照之前的保留天數。您無法在 Amazon DocumentDB 叢集上停用自動備份。

除了設定備份保留期之外，您也可以設定備份時段，也就是一天中建立自動快照的期間。

- 刪除自動快照 — 當您刪除自動快照集的叢集時，會刪除自動快照。您無法手動刪除自動快照。
- 增量 — 在備份保留期間，會記錄資料庫更新，以便有變更的遞增記錄。
- 從自動快照還原 — 您可以使用或從自動快照 AWS Management Console 還原 AWS CLI。使用從快照還原時 AWS CLI，必須在叢集可用之後分別新增執行個體。
- 共用 — 您無法共用 Amazon DocumentDB 自動化叢集快照。因應措施是，您可以藉由複製自動快照來建立手動快照，然後共用該複本。如需複製快照的詳細資訊，請參閱 [複製 Amazon DocumentDB 叢集快照](#)。如需從快照還原叢集的詳細資訊，請參閱 [從叢集快照還原](#)。

- 您可以從備份保留期間內的任何時間點還原 — 由於資料庫更新是以遞增方式記錄的，因此您可以將叢集還原至備份保留期內的任何時間點。

當您使用自動快照或從 point-in-time 還原還原時 AWS CLI，必須在叢集可用之後分別新增執行個體。

Amazon DocumentDB 手動快照具有下列主要功能：

- 視需求建立 — Amazon DocumentDB 手動快照是使用 Amazon DocumentDB 管理主控台或隨選建立的。AWS CLI
- 刪除手動快照 — 只有當您使用 Amazon DocumentDB 主控台或明確刪除手動快照時，才會刪除該快照。AWS CLI當您刪除手動快照的叢集時，並不會刪除手動快照。
- 完整備份 — 擷取手動快照時，會建立並儲存叢集資料的完整備份。
- 手動快照命名 — 您可以指定手動快照名稱。Amazon DocumentDB 不會在名稱中添加datetime戳記，因此如果您希望將其包含在名稱中，則必須添加該信息。
- 從手動快照還原 — 您可以使用主控台或從手動快照還原。AWS CLI使用從快照還原時 AWS CLI，必須在叢集可用之後分別新增執行個體。
- Service Quotas — 每個最多只能使用 100 個手動快照 AWS 區域。
- 共用 — 您可以共用手動叢集快照，這些快照可透過授權進行複製 AWS 帳戶。您可以共用加密或未加密的手動快照。如需複製快照的詳細資訊，請參閱 [複製 Amazon DocumentDB 叢集快照](#)。
- 您可以恢復到拍攝手動快照的時間 — 從手動快照還原時，會恢復到拍攝手動快照的時間。

使用從快照還原時 AWS CLI，必須在叢集可用之後分別新增執行個體。

建立手動叢集快照

您可以使用 AWS Management Console 或建立手動快照 AWS CLI。建立快照所需的時間量因資料庫的大小而異。建立快照時，您必須執行下列作業：

1. 找出要備份的叢集。
2. 為您的快照命名。您稍後可用它來進行還原。

Using the AWS Management Console

若要使用建立手動快照 AWS Management Console，您可以遵循以下任一方法。

1. 方法 1：

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇快照。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 Snapshots (快照) 頁面選擇 Create (建立)。
4. 在 Create cluster snapshot (建立叢集快照) 頁面：
 - a. 叢集 ID — 從叢集的下拉式清單中，選擇您要建立快照的叢集。
 - b. 快照識別碼 — 輸入快照的名稱。

快照命名限制條件：

- 長度為 [1—255] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- 每個 AWS 帳戶、每個區域的所有叢集 (跨 Amazon RDS、亞馬遜海王星和 Amazon DocumentDB) 必須是唯一的。

- c. 選擇建立。

2. 方法 2：

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 Clusters (叢集) 頁面，找到您要建立快照的叢集，並選擇叢集左側的按鈕。

5. 在 Create cluster snapshot (建立叢集快照) 頁面：

a. 快照識別碼 — 輸入快照的名稱。

快照命名限制條件：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- 每個 AWS 帳戶、每個區域的所有叢集 (跨 Amazon RDS、亞馬遜海王星和 Amazon DocumentDB) 必須是唯一的。

b. 選擇建立。

Using the AWS CLI

若要使用建立叢集快照 AWS CLI，請搭配下列參數使用 `create-db-cluster-snapshot` 作業。

參數

- **`--db-cluster-identifier`** – 必要。您正在建立快照的叢集名稱。此叢集必須存在並且為可用。
- **`--db-cluster-snapshot-identifier`** – 必要。您正在建立的手動快照名稱。

下列範例會為名為 `sample-cluster` 的叢集建立一個名為 `sample-cluster-snapshot` 的快照。

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster-snapshot \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

針對 Windows：

```
aws docdb create-db-cluster-snapshot ^  
  --db-cluster-identifier sample-cluster ^  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

此操作的輸出將會如下所示。


```
{
  "DBClusterSnapshot": {
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1c"
    ],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
    "DBClusterIdentifier": "sample-cluster",
    "SnapshotCreateTime": "2020-04-24T04:59:08.475Z",
    "Engine": "docdb",
    "Status": "creating",
    "Port": 0,
    "VpcId": "vpc-abc0123",
    "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
    "MasterUsername": "master-user",
    "EngineVersion": "4.0.0",
    "SnapshotType": "manual",
    "PercentProgress": 0,
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:<accountID>:cluster-snapshot:sample-cluster-snapshot"
  }
}
```

複製 Amazon DocumentDB 叢集快照

在 Amazon DocumentDB 中，您可以將手動和自動快照複製到相同帳戶內的相同 AWS 區域 或不 AWS 區域 同快照。您還可以共享由其他人擁有 AWS 帳戶 的快照 AWS 區域。但是，您無法 AWS 帳戶 在單一步驟之 AWS 區域 間複製叢集快照。這些動作必須個別執行。

作為複製的替代方法，您也可以與其他人共用手動快照 AWS 帳戶。如需詳細資訊，請參閱 [共用 Amazon DocumentDB 叢集快照](#)。

Note

Amazon DocumentDB 會根據您保留的備份和快照資料量以及保留的時間段向您收取費用。如需與 Amazon DocumentDB 備份和快照相關聯儲存的詳細資訊，請參閱 [了解備份儲存用量](#) 如需 Amazon 文件資料庫儲存的定價資訊，請參閱 [Amazon DocumentDB 定價](#)。

主題

- [複製共用快照](#)
- [複製快照 AWS 區域](#)
- [限制](#)
- [處理加密](#)
- [參數群組考量](#)
- [複製叢集快照](#)

複製共用快照

您可以複製其他人共用給您的快照 AWS 帳戶。如果您要複製已從另一個快照共用的加密快照 AWS 帳戶，您必須擁有用來 AWS KMS 加密快照的加密金鑰的存取權。

無論快照是否已加密 AWS 區域，您都只能複製相同的共用快照。如需詳細資訊，請參閱 [處理加密](#)。

複製快照 AWS 區域

當您將快照複製到與來源快照不同的快照時 AWS 區域，每個複本都是完整快照。AWS 區域 完整快照副本包含還原 Amazon DocumentDB 叢集所需的所有資料和中繼資料。

視 AWS 區域 涉及的資料量和要複製的資料量而定，跨區域快照副本可能需要數小時才能完成。在某些情況下，可能會有大量來自指定來源 AWS 區域的跨區域快照複製請求。在這些情況下，Amazon DocumentDB 可能會將來自該來源的新跨區域副本請求放入佇列中，直到某些 AWS 區域 進行中的複本完成為止。位於佇列中的複製請求不會顯示進度資訊。複製開始時才會顯示進度資訊。

限制

以下是複製快照時的一些限制：

- 如果您在目標快照變成可用之前刪除來源快照，則快照複製可能會失敗。刪除來源快照之前，請確認目標快照的狀態為 AVAILABLE。
- 對於單一目的地區域，每一帳戶最多可有 5 個快照複製請求在進行中。
- 根據涉及的區域和要複製的資料量而定，跨區域快照複製可能需要數小時才會完成。如需詳細資訊，請參閱 [複製快照 AWS 區域](#)。

處理加密

您可以複製以 AWS KMS 加密金鑰所加密的快照。如果您複製加密快照，則快照的副本也必須加密。如果您複製其中的加密快照 AWS 區域，您可以使用與原始快照相同的 AWS KMS 加密金鑰來加密副本，或者指定不同的 AWS KMS 加密金鑰。如果您跨區域複製加密的快照，則無法對副本使用與來源快照相同的 AWS KMS 加密金鑰，因為 AWS KMS 金鑰是區域特定的。相反地，您必須指定在目標 AWS 區域 n 中有效的 AWS KMS 金鑰。

在整個複製過程中來源快照仍會保持加密狀態。如需詳細資訊，請參閱 [Amazon DocumentDB 中的數據保護](#)。

Note

對於 Amazon DocumentDB 叢集快照，您無法在複製快照時加密未加密的叢集快照。

參數群組考量

跨區域複製快照時，副本不會包含原始 Amazon DocumentDB 叢集所使用的參數群組。當您還原快照以建立新叢集時，該叢集會取得在 AWS 區域 其中建立快照的預設參數群組。若要為新叢集提供與原始叢集相同的參數，您必須執行下列動作：

1. 在目的地中 AWS 區域，使用 [與原始叢集相同的設定建立 Amazon DocumentDB 叢集參數群組](#)。如果一個已經存在於新的 AWS 區域，你可以使用那個。
2. 在目標中還原快照之後 AWS 區域，請修改新的 Amazon DocumentDB 叢集，並從上一步中新增新的或現有的參數群組。如需詳細資訊，請參閱 [修改 Amazon DocumentDB 叢集](#)。

複製叢集快照

您可以使用 AWS Management Console 或複製 Amazon DocumentDB 叢集 AWS CLI，如下所示。

Using the AWS Management Console

若要使用複製叢集快照 AWS Management Console，請完成下列步驟。此程序適用於複製相同或跨區域的加密 AWS 區域 或未加密叢集快照。

1. [登入 AWS Management Console](#)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在功能窗格中，選擇 [快照]，然後選擇您要複製的快照左側的按鈕。

i Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 從 Actions (動作) 功能表中，選擇 Copy (複製)。
4. 在產生的 [建立叢集快照複本] 頁面中，完成 [設定] 區段。
 - a. 目的地區域 — 選用。若要將叢集快照複製到其他快照 AWS 區域，請 AWS 區域 為 [目的地區域] 選擇該快照。
 - b. 新快照識別碼 — 輸入新快照的名稱。

目標快照命名限制條件：

- 不能是現有快照的名稱。
- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- Amazon RDS、Neptune 和亞馬 Amazon DocumentDB 中每 AWS 帳戶個區域的所有叢集必須是唯一的。

- c. 複製標籤 — 若要將來源快照上的任何標籤複製到快照副本，請選擇 [複製標籤]。
5. 完成 Encryption-at-rest 部分。
 - a. 靜態加密 — 如果您的快照未加密，您將無法使用這些選項，因為您無法從未加密的快照建立加密副本。如果您的快照已加密，您可以變更靜態加密期間 AWS KMS key 使用的快照。

如需加密快照複本的詳細資訊，請參閱[複製叢集快照加密](#)。

如需靜態加密的詳細資訊，請參閱[加密 Amazon DocumentDB 中的靜態數據](#)。

- b. 主要金鑰 — 從下拉式清單中選擇下列其中一項：
 - (預設值) aws/rds — 帳號和 AWS KMS 金鑰識別碼會列在此選項之後。
 - < some-key-name > — 如果您建立了金鑰，就會列出該金鑰，並可供您選擇。
 - 輸入金鑰 ARN — 在 ARN 方塊中，輸入金鑰的 Amazon 資源名稱 (ARN)。AWS KMS ARN 的格式是 `arn:aws:kms:<region>:<accountID>:key/<key-id>`。

- 若要製作所選快照的複本，請選擇 Copy snapshot (複製快照)。或者，您可以選擇 [取消]，不建立快照的複本。

Using the AWS CLI

若要使用複製未加密的叢集快照 AWS CLI，請搭配下列參數使用此 `copy-db-cluster-snapshot` 作業。如果您要將快照複製到另一個快照 AWS 區域，請執行快照 AWS 區域 要複製到的目標中的命令。

- **--source-db-cluster-snapshot-identifier** – 必要。要製作複本之叢集快照的識別碼。叢集快照必須存在且位於可用狀態。如果您要將快照複製到另一個快照 AWS 區域，則此識別碼必須為來源 AWS 區域的 ARN 格式。此參數不區分大小寫。
- **--target-db-cluster-snapshot-identifier** – 必要。要從來源叢集快照建立之新叢集快照的識別碼。此參數不區分大小寫。

目標快照命名限制條件：

- 不能是現有快照的名稱。
- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- Amazon RDS、Neptune 和亞馬 Amazon DocumentDB 中每 AWS 帳戶個區域的所有叢集必須是唯一的。
- **--source-region**— 如果您要將快照複製到另一個快照 AWS 區域，請指 AWS 區域 定要從中複製加密的叢集快照。

如果您要將快照複製到另一個 AWS 區域 快照，但未指定 `--source-region`，則必須改為指定 `pre-signed-url` 選項。該 `pre-signed-url` 值必須是包含簽名版本 4 簽署請求的 URL，以便在複製叢集快照 AWS 區域 的來源中呼叫 `CopyDBClusterSnapshot` 動作。若要進一步了解 `pre-signed-url`，請參閱 [CopyDB ClusterSnapshot](#)。

- **--kms-key-id**— 用來加密叢集快照副本的金鑰 KMS 金鑰識別碼。

如果要將加密的叢集快照複製到另一個快照 AWS 區域，則需要此參數。您必須指定目的地的 KMS 金鑰 AWS 區域。

如果您要複製相同的加密叢集快照 AWS 區域，則 AWS KMS key 參數是選用的。叢集快照的副本會使用與來源叢集快照相同的 AWS KMS 金鑰加密。如果您想要指定用來 AWS KMS 加密副本的新加密金鑰，可以使用此參數進行加密。

- **--copy-tags**-可選。要複製的標籤和值。

若要在進行複製作業之後取消複製作業，您可以刪除該叢集快照在複製狀態TargetDBClusterSnapshotIdentifier時識別的目標叢集快照，`--target-db-cluster-snapshot-identifier`或在該叢集快照處於複製狀態時刪除。

Example

範例 1：將未加密的快照複製到相同區域

下列 AWS CLI 範例會建立與來源快照相同 AWS 區域 同的sample-cluster-snapshot名稱sample-cluster-snapshot-copy複本。建立副本時，原始快照的所有標籤會複製到快照副本。

針對 Linux、macOS 或 Unix：

```
aws docdb copy-db-cluster-snapshot \  
  --source-db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \  
  --copy-tags
```

針對 Windows：

```
aws docdb copy-db-cluster-snapshot ^  
  --source-db-cluster-snapshot-identifier sample-cluster-snapshot ^  
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy ^  
  --copy-tags
```

此操作的輸出將會如下所示。

```
{  
  "DBClusterSnapshot": {  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1b",  
      "us-east-1c"  
    ],  
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",  
    "DBClusterIdentifier": "sample-cluster",  
    "SnapshotCreateTime": "2020-03-27T08:40:24.805Z",  
    "Engine": "docdb",  
    "Status": "copying",
```

```

    "Port": 0,
    "VpcId": "vpc-abcd0123",
    "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
    "MasterUsername": "master-user",
    "EngineVersion": "4.0.0",
    "SnapshotType": "manual",
    "PercentProgress": 0,
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/sample-key-id",
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
    "SourceDBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot"
  }
}

```

Example

範例 2：複製未加密的快照 AWS 區域

下列 AWS CLI 範例會建立具有 ARN `arn:aws:rds:us-east-1:123456789012:cluster-snapshot:sample-cluster-snapshot` 的 `sample-cluster-snapshot` 複本。此副本命名為 `sample-cluster-snapshot-copy` 並且 AWS 區域 在其中執行命令。

若為 Linux、macOS 或 Unix：

```

aws docdb copy-db-cluster-snapshot \
  --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:sample-cluster-snapshot \
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy

```

針對 Windows：

```

aws docdb copy-db-cluster-snapshot ^
  --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:sample-cluster-snapshot ^
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy

```

此操作的輸出將會如下所示。

```

{
  "DBClusterSnapshot": {

```

```

    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1c"
    ],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
    "DBClusterIdentifier": "sample-cluster",
    "SnapshotCreateTime": "2020-04-29T16:45:51.239Z",
    "Engine": "docdb",
    "AllocatedStorage": 0,
    "Status": "copying",
    "Port": 0,
    "VpcId": "vpc-abc0123",
    "ClusterCreateTime": "2020-04-28T16:43:00.294Z",
    "MasterUsername": "master-user",
    "EngineVersion": "4.0.0",
    "LicenseModel": "docdb",
    "SnapshotType": "manual",
    "PercentProgress": 0,
    "StorageEncrypted": false,
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
    "SourceDBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot",
  }
}

```

Example

範例 3：複製加密快照 AWS 區域

下列 AWS CLI 範例會建立 `sample-cluster-snapshot` 從 `us-west-2` 區域到 `us-east-1` 區域的複本。此命令在 `us-east-1` 區域中被調用。

若為 Linux、macOS 或 Unix：

```

aws docdb copy-db-cluster-snapshot \
  --source-db-cluster-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:cluster-snapshot:sample-cluster-snapshot \
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
  --source-region us-west-2 \
  --kms-key-id sample-us-east-1-key

```


針對 Windows :

```
aws docdb copy-db-cluster-snapshot ^
  --source-db-cluster-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:cluster-snapshot:sample-cluster-snapshot ^
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy ^
  --source-region us-west-2 ^
  --kms-key-id sample-us-east-1-key
```

此操作的輸出將會如下所示。

```
{
  "DBClusterSnapshot": {
    "AvailabilityZones": [],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
    "DBClusterIdentifier": "ayhu-xrsc-test-ap-southeast-1-small-cluster-kms",
    "SnapshotCreateTime": "2020-04-29T16:45:53.159Z",
    "Engine": "docdb",
    "AllocatedStorage": 0,
    "Status": "copying",
    "Port": 0,
    "ClusterCreateTime": "2020-04-28T16:43:07.129Z",
    "MasterUsername": "chimera",
    "EngineVersion": "4.0.0",
    "LicenseModel": "docdb",
    "SnapshotType": "manual",
    "PercentProgress": 0,
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/sample-key-id",
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
    "SourceDBClusterSnapshotArn": "arn:aws:rds:us-west-2:111122223333:cluster-
snapshot:sample-cluster-snapshot",
  }
}
```

Note

如需加密快照複本的詳細資訊，請參閱[複製叢集快照加密](#)。

如需靜態加密的詳細資訊，請參閱[加密 Amazon DocumentDB 中的靜態數據](#)。

共用 Amazon DocumentDB 叢集快照

在 Amazon DocumentDB 中，您可以共用手動叢集快照，這些快照可由授權進行複製。AWS 帳戶您可以共用加密或未加密的手動快照。共用未加密的快照時，授權 AWS 帳戶可以直接從快照還原叢集，而不是複製叢集並從該快照還原叢集。然而，您無法從已共用且加密的快照還原叢集。不過，您可以複製叢集，再從該複本還原叢集。如需複製快照的詳細資訊，請參閱 [複製 Amazon DocumentDB 叢集快照](#)。

Note

您無法共用 Amazon DocumentDB 自動化叢集快照。因應措施是，您可以藉由複製自動快照來建立手動快照，然後共用該複本。如需複製快照的詳細資訊，請參閱 [複製 Amazon DocumentDB 叢集快照](#)。如需從快照還原叢集的詳細資訊，請參閱 [從叢集快照還原](#)。

您最多可與 20 個其他人共用手動快照 AWS 帳戶。您也可以將未加密的手動快照以公有形式共用，讓所有帳戶都可使用此快照。將快照以公有形式共用時，請確保不要在任何公有快照中包含您的私人資訊。

與其他人共用手動快照 AWS 帳戶，並使用 AWS CLI 或 Amazon DocumentDB API 從共用快照還原叢集時，您必須指定共用快照的 Amazon 資源名稱 (ARN) 做為快照識別碼。

共用加密快照

共用加密快照有下列限制：

- 您無法將加密快照以公有形式共用。
- 您無法共用已使用共用快照之帳戶的預設 AWS KMS 加密金鑰加密的快照。

請按照以下步驟共用加密的快照。

1. 與任何您希望能夠存取快照的帳戶共用用於加密快照的 AWS Key Management Service (AWS KMS) 加密金鑰。

您可以將其他 AWS 帳戶新增至金 AWS KMS 鑰策略，與其他帳戶共用加密金 AWS KMS 鑰。如需更新金鑰原則的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [使用 AWS KMS 中的金鑰原則](#)。有關建立金鑰政策的範例，請參閱本主題稍後的 [建立 IAM 政策以允許複製加密快照](#)。

2. 使用 AWS CLI(如下所示) 與其他帳戶共用加密快照。

允許存取加 AWS KMS 密金鑰

AWS 帳戶 若要複製從您的帳戶共用的加密快照，您共用快照的帳戶必須能夠存取加密快照的 AWS KMS 金鑰。若要允許其他帳戶存取 AWS KMS 金鑰，請使用您要共用的帳戶的 ARN 來更新 AWS KMS 金鑰的金鑰原則，做為 AWS KMS 金鑰原則中的主要使用者。然後允許 `kms:CreateGrant` 動作。

授予帳戶存取 AWS KMS 加密金鑰之後，若要複製加密快照，該帳戶必須建立 AWS Identity and Access Management (IAM) 使用者 (如果該使用者還沒有)。此外，該帳戶還必須將 IAM 政策附加到該 IAM 使用者，以允許使用者使用您的 AWS KMS 金鑰複製加密快照。該帳戶必須是 IAM 使用者，且由於 AWS KMS 安全限制，不能是 root AWS 帳戶 身分。

在下列金鑰原則範例中，使用者 123451234512 是加密金鑰的擁有者。AWS KMS 使用者 123456789012 是共用金鑰的帳戶。此更新的金鑰政策可讓帳戶存取 AWS KMS 金鑰。它會將使用者 123456789012 的根 AWS 帳戶 識別的 ARN 納入為原則的主參與者，並允許動作來達到此目的。`kms:CreateGrant`

```
{
  "Id": "key-policy-1",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::123451234512:user/KeyUser",
        "arn:aws:iam::123456789012:root"
      ]},
      "Action": [
        "kms:CreateGrant",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow attachment of persistent resources",
```

```

    "Effect": "Allow",
    "Principal": {"AWS": [
      "arn:aws:iam::123451234512:user/KeyUser",
      "arn:aws:iam::123456789012:root"
    ]},
    "Action": [
      "kms:CreateGrant",
      "kms:ListGrants",
      "kms:RevokeGrant"
    ],
    "Resource": "*",
    "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
  }
]
}

```

建立 IAM 政策以允許複製加密快照

當外部可以存取您的 AWS KMS 金鑰時，該帳戶的擁 AWS 帳戶 有者可以建立政策，允許為該帳戶建立的 IAM 使用者複製使用該 AWS KMS 金鑰加密的加密快照。

下列範例顯示可以附加至身分與存取權管理使用者的政策，適用於 AWS 帳戶 123456789012。該政策可讓 IAM 使用者從帳戶 123451234512 複製共用快照，該帳戶已使用 us-west-2 區域中的金鑰加密。AWS KMS c989c1dd-a3f2-4a5d-8d96-e793d082ab26

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUseOfTheKey",
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:RetireGrant"
      ],
      "Resource": ["arn:aws:kms:us-west-2:123451234512:key/c989c1dd-a3f2-4a5d-8d96-e793d082ab26"]
    },
  ],
}

```

```

    {
      "Sid": "AllowAttachmentOfPersistentResources",
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": ["arn:aws:kms:us-west-2:123451234512:key/c989c1dd-
a3f2-4a5d-8d96-e793d082ab26"],
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": true
        }
      }
    }
  ]
}

```

如需更新金鑰原則的詳細資訊，請參閱AWS Key Management Service 開發人員指南 [AWS KMS中的使用金鑰政策](#)。

共享快照

若要共用快照，請使用 Amazon DocumentDB `modify-db-snapshot-attribute` 作業。使用 `--values-to-add` 參數可新增授權可還原手動快照的 ID 清單。AWS 帳戶

下列範例允許兩個 AWS 帳戶 識別碼 (123451234512 和 123456789012) 還原名為的快照集。manual-snapshot1這還會移除 all 屬性值，以將快照標記為私有。

若為 Linux、macOS 或 Unix：

```

aws docdb modify-db-cluster-snapshot-attribute \
  --db-cluster-snapshot-identifier sample-cluster-snapshot \
  --attribute-name restore \
  --values-to-add '["123451234512","123456789012"]'

```

針對 Windows：

```

aws docdb modify-db-cluster-snapshot-attribute ^
  --db-cluster-snapshot-identifier sample-cluster-snapshot ^

```

```
--attribute-name restore ^
--values-to-add '["123451234512","123456789012"]'
```

此操作的輸出將會如下所示。

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123451234512",
          "123456789012"
        ]
      }
    ]
  }
}
```

若要從清單中移除 AWS 帳戶 識別碼，請使用 `--values-to-remove` 參數。下列範例會防止 AWS 帳戶 識別碼 123456789012 還原快照集。

若為 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster-snapshot-attribute \
  --db-cluster-snapshot-identifier sample-cluster-snapshot \
  --attribute-name restore \
  --values-to-remove '["123456789012"]'
```

針對 Windows：

```
aws docdb modify-db-cluster-snapshot-attribute ^
  --db-cluster-snapshot-identifier sample-cluster-snapshot ^
  --attribute-name restore ^
  --values-to-remove '["123456789012"]'
```

此操作的輸出將會如下所示。

```
{
```

```
"DBClusterSnapshotAttributesResult": {
  "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
  "DBClusterSnapshotAttributes": [
    {
      "AttributeName": "restore",
      "AttributeValues": [
        "123451234512"
      ]
    }
  ]
}
```

從叢集快照還原

Amazon DocumentDB (與 MongoDB 相容性) 會建立儲存磁碟區的叢集快照。您可以藉由從叢集快照進行還原來建立新叢集。還原叢集時，您需提供做為還原來源的叢集快照名稱，以及該還原所建立的新叢集名稱。您無法從快照還原到現有叢集，因為新叢集會在您還原時建立。

當您從叢集快照還原叢集時：

- 這個動作只會還原叢集，而不會還原該叢集的執行個體。您必須呼叫 `create-db-instance` 動作，為還原的叢集建立執行個體，並且在 `--db-cluster-identifier` 中指定所還原叢集的識別碼。只有當叢集可用時，才可以建立執行個體。
- 您無法將已加密的快照還原至未加密的叢集。不過，您可以指定 AWS KMS 金鑰，將未加密的快照還原至加密的叢集。
- 若要從加密的快照還原叢集，您必須擁有 AWS KMS 金鑰的存取權。


Note

您無法將 3.6 叢集還原到 4.0 叢集，但可以從一個叢集版本移轉到另一個叢集版本。如需詳細資訊，請前往 [遷移到 Amazon DocumentDB](#)。

Using the AWS Management Console


下列程序說明如何使用 Amazon DocumentDB 管理主控台從叢集快照還原 Amazon DocumentDB 叢集。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)，找到您要用於還原至叢集的快照，並選擇其左側的按鈕。

 Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在操作功能表上，選擇 Restore (還原)。
4. 在 Restore snapshot (還原快照) 頁面，完成 Configuration (組態設定) 區段。
 - a. 叢集識別碼 — 新叢集的名稱。您可以接受 Amazon DocumentDB 提供的名稱或鍵入您喜歡的名稱。Amazon 文件資料庫提供的名稱格式為 docdb-加上 UTC 時間戳記，例如。docdb-yyyy-mm-dd-hh-mm-ss
 - b. 執行個體類別 — 新叢集的執行個體類別。您可以接受預設執行個體類別，或從下拉式清單中選擇執行個體類別。
 - c. 執行個體數目 — 您要使用此叢集建立的執行處理數目。您可以接受預設的 3 個執行個體 (1 個主要讀取/寫入和 2 個唯讀複本)，或從下拉式清單中選擇執行個體數目。
5. 針對叢集儲存裝置組態，請選擇儲存選項。

 Note

Amazon DocumentDB I/O 優化的存儲配置僅適用於 Amazon DocumentDB 5.0 引擎版本。

6. 如果您滿意此叢集組態，請選擇 Restore cluster (還原叢集)，等待叢集還原。
7. 如果您想要變更某些組態 (例如指定非預設 Amazon VPC 或安全群組)，請選擇頁面左下角的 [顯示進階設定]，然後繼續執行下列步驟。
 - a. 完成 Network settings (網路設定) 區段。
 - V@@rtual Private Cloud (VPC) (VPC) — 接受目前的 VPC，或從下拉式清單中選擇 VPC。
 - 子網路群組 — 接受 default 子網路群組，或從下拉式清單中選擇一個子網路群組。
 - VPC 安全群組 — 接受 default (VPC) 安全群組，或從清單中選擇一個群組。

- b. 填寫 Cluster options (叢集選項) 區段。
 - 資料庫連接埠 — 接受預設連接埠 27017，或使用向上鍵或向下鍵來設定您要用於應用程式連線的連接埠。
 - c. 填寫 Encryption (加密) 區段。
 - 靜態加密 — 如果您的快照已加密，則您無法使用這些選項。如果未加密，您可以選擇以下其中一項：
 - 若要加密叢集的所有資料，請選擇 [啟用] encryption-at-rest。如果選擇此選項，您必須指定 Master key (主要金鑰)。
 - 若要不加密叢集的資料，請選擇 [停用] encryption-at-rest。如果選擇此選項，您會在加密區段完成。
 - 主金鑰 — 從下拉式清單中選擇下列其中一項：
 - (預設值) aws/rds — 帳號和 AWS KMS 金鑰識別碼會列在此選項之後。
 - 客戶管理金鑰 — 只有在您在 AWS Identity and Access Management (IAM) 主控台中建立 IAM 加密金鑰時，才能使用此選項。您可以選擇金鑰加密您的叢集。
 - 輸入金鑰 ARN — 在 ARN 方塊中，輸入金鑰的 Amazon 資源名稱 (ARN)。AWS KMS ARN 的格式是 `arn:aws:kms:<region>:<accountID>:key/<key-id>`。
 - d. 完成 Log exports (日誌匯出) 區段。
 - 選取要發佈至的防護記錄類型 CloudWatch — 選擇下列其中一項：
 - 啟用 — 可讓您的叢集將 DDL 日誌記錄匯出至 Amazon CloudWatch 日誌。
 - 停用 — 防止叢集將 DDL 日誌匯出至 Amazon CloudWatch 日誌。Disabled (已停用) 是預設值。
 - IAM 角色 — 從清單中選擇 RDS 服務連結角色。
 - e. 完成 Tags (標籤) 區段。
 - 新增標籤 — 在 [金鑰] 方塊中，輸入叢集標籤的名稱。在 Value (值) 方塊中，選擇性地輸入標籤值。標籤可與 AWS Identity and Access Management (IAM) 政策搭配使用，以管理對 Amazon DocumentDB 資源的存取，以及控制哪些動作可套用至資源。
 - f. 完成 Deletion protection (刪除保護) 區段。
 - 啟用刪除保護 — 防止叢集遭到意外刪除。啟用此選項後，您無法刪除叢集。
8. 選擇 Restore cluster (還原叢集)。

Using the AWS CLI

若要使用從快照還原叢集 AWS CLI，請使用具有下列參數的 `restore-db-cluster-from-snapshot` 作業。如需詳細資訊，請參閱 [RestoreDBClusterFromSnapshot](#)。

- **--db-cluster-identifier** – 必要。這項操作會建立的叢集名稱。執行此操作之前，該叢集名稱不可以存在。

叢集命名限制條件：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- Amazon RDS、Neptune 和亞馬 Amazon DocumentDB 中每 AWS 帳戶個區域的所有叢集必須是唯一的。
- **--snapshot-identifier** – 必要。用來還原的快照名稱。此名稱的快照必須存在且位於可用狀態。
- **--engine** – 必要。必須為 `docdb`。
- **--storage-type standard | iopt1**-可選。預設：`standard`。
- **--kms-key-id**-可選。從未加 AWS KMS 密的快照還原時還原加密快照或加密叢集時所使用的金鑰識別碼 ARN。提供 AWS KMS 金鑰識別碼會導致還原的叢集使用 AWS KMS 金鑰加密，無論快照是否已加密。

`--kms-key-id` 的格式是 `arn:aws:kms:<region>:<accountID>:key/<key-id>`。如果您未指定 `--kms-key-id` 參數的值，會發生以下情況：

- 如果中的快照 `--snapshot-identifier` 已加密，則還原的叢集會使用用於加密快照的相同 AWS KMS 金鑰來加密。
- 如果 `--snapshot-identifier` 中的快照未加密，則不會加密已還原的叢集。

若為 Linux、macOS 或 Unix：

```
aws docdb restore-db-cluster-from-snapshot \  
  --db-cluster-identifier sample-cluster-restore \  
  --snapshot-identifier sample-cluster-snapshot \  
  --engine docdb \  
  --kms-key-id arn:aws:kms:us-east-1:123456789012:key/SAMPLE-KMS-KEY-ID
```

針對 Windows :

```
aws docdb restore-db-cluster-from-snapshot ^
  --db-cluster-identifier sample-cluster-restore ^
  --snapshot-identifier sample-cluster-snapshot ^
  --engine docdb ^
  --kms-key-id arn:aws:kms:us-east-1:123456789012:key/SAMPLE-KMS-KEY-ID
```

此操作的輸出將會如下所示。

```
{
  "DBCluster": {
    "AvailabilityZones": [
      "us-east-1c",
      "us-east-1b",
      "us-east-1a"
    ],
    "BackupRetentionPeriod": 1,
    "DBClusterIdentifier": "sample-cluster-restore",
    "DBClusterParameterGroup": "default.docdb4.0",
    "DBSubnetGroup": "default",
    "Status": "creating",
    "Endpoint": "sample-cluster-restore.cluster-node.us-
east-1.docdb.amazonaws.com",
    "ReaderEndpoint": "sample-cluster-restore.cluster-node.us-
east-1.docdb.amazonaws.com",
    "MultiAZ": false,
    "Engine": "docdb",
    "EngineVersion": "4.0.0",
    "Port": 27017,
    "MasterUsername": "<master-user>",
    "PreferredBackupWindow": "02:00-02:30",
    "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
    "DBClusterMembers": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-abcdefgh",
        "Status": "active"
      }
    ],
    "HostedZoneId": "ABCDEFGHIJKLM",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/<sample-key-id>",
```

```

    "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNQPQRSTUVWXYZ",
    "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-restore",
    "AssociatedRoles": [],
    "ClusterCreateTime": "2020-04-01T01:43:40.871Z",
    "DeletionProtection": true
  }
}

```

當叢集狀態可用時，為叢集至少建立一個執行個體。

若為 Linux、macOS 或 Unix：

```

aws docdb create-db-instance \
  --db-cluster-identifier sample-cluster-restore \
  --db-instance-identifier sample-cluster-restore-instance \
  --availability-zone us-east-1b \
  --promotion-tier 2 \
  --db-instance-class db.r5.large \
  --engine docdb

```

針對 Windows：

```

aws docdb create-db-instance ^
  --db-cluster-identifier sample-cluster-restore ^
  --db-instance-identifier sample-cluster-restore-instance ^
  --availability-zone us-east-1b ^
  --promotion-tier 2 ^
  --db-instance-class db.r5.large ^
  --engine docdb

```

此操作的輸出將會如下所示。

```

{
  "DBInstance": {
    "DBInstanceIdentifier": "sample-cluster-restore-instance",
    "DBInstanceClass": "db.r5.large",
    "Engine": "docdb",
    "DBInstanceStatus": "creating",
    "PreferredBackupWindow": "02:00-02:30",
    "BackupRetentionPeriod": 1,
    "VpcSecurityGroups": [
      {

```

```
        "VpcSecurityGroupId": "sg-abcdefgh",
        "Status": "active"
    }
],
"AvailabilityZone": "us-west-2b",
"DBSubnetGroup": {
    "DBSubnetGroupName": "default",
    "DBSubnetGroupDescription": "default",
    "VpcId": "vpc-6242c31a",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
        {
            "SubnetIdentifier": "subnet-abcdefgh",
            "SubnetAvailabilityZone": {
                "Name": "us-west-2a"
            },
            "SubnetStatus": "Active"
        },
        {
            ...
        }
    ]
},
"PreferredMaintenanceWindow": "fri:09:43-fri:10:13",
"PendingModifiedValues": {},
"EngineVersion": "4.0.0",
"AutoMinorVersionUpgrade": true,
"PubliclyAccessible": false,
"DBClusterIdentifier": "sample-cluster-restore",
"StorageEncrypted": true,
"KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/<sample-key-id>",
"DbiResourceId": "db-ABCDEFGHIJKLMNQRSTUWXYZ",
"CACertificateIdentifier": "rds-ca-2019",
"PromotionTier": 2,
"DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-cluster-
restore-instance"
}
}
```

時間點還原

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 將叢集還原到叢集備份保留期內的任何時間點。

Note

您無法將 3.6 叢集 point-in-time 還原至 4.0 叢集，但可以從一個叢集版本移轉到另一個叢集版本。如需詳細資訊，請前往 [遷移到 Amazon DocumentDB](#)。

將叢集還原至某個時間點時，請謹記下列事項。

- 除非是以預設的參數群組建立新叢集，否則會以與來源叢集相同的組態建立新叢集。若要將新叢集的參數群組設為來源叢集的參數群組，請在叢集狀態變為「可用」後修改它。如需修改叢集的詳細資訊，請參閱 [修改 Amazon DocumentDB 叢集](#)。

Using the AWS Management Console

您可以使用完成下列作業，將叢集還原到備份保留期 point-in-time 內的 AWS Management Console。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。在叢集清單中，選擇您要還原之叢集左側的按鈕。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。


3. 在 Actions (動作) 功能表中，選擇 Restore to point in time (還原至時間點)。
4. 填妥 Restore time (還原時間) 部分，這裡指定還原的日期和時間。
 - a. 還原日期 — 選擇或輸入介於「最早」還原時間與「最新」還原時間之間的日期。
 - b. 還原時間 — 選擇或輸入介於「最早」還原時間與「最新」還原時間之間的小時、分鐘和秒數。
5. 填妥 Configuration (組態) 部分。

- a. 叢集識別碼 — 接受預設識別碼，或輸入您偏好的識別碼。

叢集命名限制條件：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- Amazon RDS、Neptune 和亞馬 Amazon DocumentDB 中每 AWS 帳戶個區域的所有叢集必須是唯一的。

- b. 執行個體類別 — 從下拉式清單中，選擇叢集執行個體所需的執行個體類別。
 - c. 執行個體數量 — 從下拉式清單中，選擇您要在還原叢集時建立的執行處理數目。
6. 針對叢集儲存裝置組態，請選擇儲存選項。

 Note

Amazon DocumentDB I/O 優化的存儲配置僅適用於 Amazon DocumentDB 5.0 引擎版本。

7. 選用。若要設定網路設定、叢集選項並啟用日誌匯出，請選擇 Show advanced settings (顯示進階設定)，然後完成下列區段。否則，請繼續至下一個步驟。
 - Network settings (網路設定)
 1. Virtual Private Cloud (VPC) (VPC) — 從下拉式清單中，選擇要用於此叢集的 VPC。
 2. 子網路群組 — 從下拉式清單中，選擇此叢集的子網路群組。
 3. VPC 安全群組 — 從下拉式清單中，選擇此叢集的 VPC 安全性群組。
 - Cluster options (叢集選項)
 1. 連接埠 — 接受預設連接埠 (27017)，或使用向上鍵和向下箭頭設定與此叢集通訊的連接埠。
 - Log exports (日誌匯出)

1. 稽核日誌 — 選取此選項即可將稽核日誌匯出至 Amazon CloudWatch 日誌。如果您選取此選項，則必須在叢集的自訂參數群組中啟用 `audit_logs`。如需詳細資訊，請參閱 [稽核 Amazon DocumentDB 事件](#)。
2. 效能分析工具日誌 — 選取此選項可讓您將作業效能分析工具日誌匯出至 Amazon 日誌。CloudWatch 如果您選取此選項，則還必須在叢集的自訂參數群組中修改下列參數：
 - `profiler`— 設定為 `enabled`。
 - `profiler_threshold_ms`— 設定為值以設 `[0-INT_MAX]` 定效能分析作業的臨界值。
 - `profiler_sampling_rate`— 設定為一個值，`[0.0-1.0]` 以設定設定檔的慢速作業百分比。

如需詳細資訊，請參閱 [分析亞馬遜文檔數據庫操作](#)。

3. 效能分析工具日誌 — 將效能分析工具日誌匯出到 Amazon CloudWatch
4. IAM 角色 — 從下拉式清單中選擇 RDS 服務連結角色。
 - Tags (標籤)
 1. 新增標籤 — 在 [金鑰] 方塊中，輸入叢集標籤的名稱。在 Value (值) 方塊中，選擇性地輸入標籤值。標籤可與 AWS Identity and Access Management (IAM) 政策搭配使用，以管理對 Amazon DocumentDB 資源的存取，以及控制哪些動作可套用至資源。
 - 刪除保護
 1. 啟用刪除保護 — 防止叢集遭到意外刪除。啟用此選項後，您無法刪除叢集。
8. 若要還原叢集，請選擇 Create cluster (建立叢集)。或者，您可以選擇 Cancel (取消) 以取消操作。

Using the AWS CLI

若要將叢集還原至快照備份保留期內的某個時間點，請搭配下列參數使用 `restore-db-cluster-to-point-in-time` 操作。

- **`--db-cluster-identifier`**— 必要。要建立之新叢集的名稱。操作前，此叢集不可以存在。參數值必須滿足下列限制條件。

叢集命名限制條件：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

- Amazon RDS、Neptune 和亞馬 Amazon DocumentDB 中每 AWS 帳戶個區域的所有叢集必須是唯一的。
- **--restore-to-time**— 將叢集還原至的 UTC 日期和時間。例如 2018-06-07T23:45:00Z。
時間限制：
 - 必須在叢集的最近一次可還原時間之前。
 - 如果未提供 **--use-latest-restorable-time** 參數，則必須指定。
 - 如果 **--use-latest-restorable-time** 參數為 true，則無法指定。
 - 如果 **--restore-type** 參數值為 copy-on-write，則無法指定。
- **--source-db-cluster-identifier**— 要從中還原的來源叢集名稱。此叢集必須存在並且為可用。
- **--use-latest-restorable-time**或 **--no-use-latest-restorable-time** — 是否還原到最新的可還原備份時間。如果未提供 **--restore-to-time** 參數，則無法指定。
- **--storage-type standard | iopt1**-可選。預設：standard。

AWS CLI 作業 `restore-db-cluster-to-point-in-time` 只會還原叢集，而不會還原該叢集的執行個體。您必須呼叫 `create-db-instance` 操作，為還原的叢集建立執行個體，並且在 `--db-cluster-identifier` 中指定所還原叢集的識別碼。只有在 `restore-db-cluster-to-point-in-time` 操作完成後，且還原的叢集為可用時，您才能建立執行個體。

Example

以下範例會從快照 `sample-cluster-snapshot` 的最近一次可還原時間建立 `sample-cluster-restored`。

若為 Linux、macOS 或 Unix：

```
aws docdb restore-db-cluster-to-point-in-time \  
  --db-cluster-identifier sample-cluster-restored \  
  --source-db-cluster-identifier sample-cluster-snapshot \  
  --use-latest-restorable-time
```

針對 Windows：

```
aws docdb restore-db-cluster-to-point-in-time ^  
  --db-cluster-identifier sample-cluster-restored ^  
  --source-db-cluster-identifier sample-cluster-snapshot ^
```

```
--use-latest-restorable-time
```

Example

以下範例會從快照 `sample-cluster-snapshot` 的 `sample-cluster` 備份保留期間中的 2018 年 12 月 11 日 03:15 (UTC) 建立 `sample-cluster-restored`。

若為 Linux、macOS 或 Unix：

```
aws docdb restore-db-cluster-to-point-in-time \  
  --db-cluster-identifier sample-cluster-restore \  
  --source-db-cluster-identifier sample-cluster \  
  --restore-to-time 2020-05-12T03:15:00Z
```

針對 Windows：

```
aws docdb restore-db-cluster-to-point-in-time ^  
  --db-cluster-identifier sample-cluster-restore ^  
  --source-db-cluster-identifier sample-cluster ^  
  --restore-to-time 2020-05-12T03:15:00Z
```

此操作的輸出將會如下所示。

```
{  
  "DBCluster": {  
    "AvailabilityZones": [  
      "us-east-1c",  
      "us-west-2b",  
      "us-west-2a"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DBClusterIdentifier": "sample-cluster-restored",  
    "DBClusterParameterGroup": "sample-parameter-group",  
    "DBSubnetGroup": "default",  
    "Status": "creating",  
    "Endpoint": "sample-cluster-restored.node.us-east-1.docdb.amazonaws.com",  
    "ReaderEndpoint": "sample-cluster-restored.node.us-  
east-1.docdb.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "docdb",  
    "EngineVersion": "4.0.0",
```

```
"Port": 27017,
"MasterUsername": "master-user",
"PreferredBackupWindow": "02:00-02:30",
"PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
"DBClusterMembers": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-abc0123",
    "Status": "active"
  }
],
"HostedZoneId": "ABCDEFGHIJKLM",
"StorageEncrypted": true,
"KmsKeyId": "arn:aws:kms:us-east-1:<accountID^>:key/sample-key",
"DbClusterResourceId": "cluster-ABCDEFGHIJKLMNQRSTUWXYZ",
"DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-restored",
"AssociatedRoles": [],
"ClusterCreateTime": "2020-04-24T20:14:36.713Z",
"DeletionProtection": false
}
}
```

刪除叢集快照

手動快照是完整備份，只有在您使用 AWS Management Console 或手動刪除快照時才會刪除 AWS CLI。您無法手動刪除自動快照，只有在快照保留期間結束或您刪除快照叢集時，才會刪除自動快照。

Using the AWS Management Console

若要使用刪除手動叢集快照 AWS Management Console，請完成下列步驟。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇快照。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示

(☰

)。

3. 在快照清單中，找到您要刪除的快照，並選擇左側的按鈕。快照的類型必須為手動。
 1. 您可以檢查快照是否列在「類型」欄中manual或automatic下方，以確認快照的類型是手動的。
4. 在 Actions (動作) 選單中，選擇 Delete (刪除)。如果 Delete (刪除) 選項無法使用，表示您可能選擇了自動快照。
5. 在刪除確認畫面上，選擇 Delete (刪除) 來刪除快照。如要保留快照，請選擇 Cancel (取消)。

Using the AWS CLI

Amazon DocumentDB 手動叢集快照是完整備份，您可以使用 AWS CLI 您無法手動刪除自動快照。

若要使用刪除手動叢集快照 AWS CLI，請搭配下列參數使用此 `delete-db-cluster-snapshot` 作業。

參數

- **`--db-cluster-snapshot-identifier`** – 必要。要刪除的手動快照名稱。

以下範例會刪除 `sample-cluster-snapshot` 叢集快照。

若為 Linux、macOS 或 Unix：

```
aws docdb delete-db-cluster-snapshot \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

針對 Windows：

```
aws docdb delete-db-cluster-snapshot ^  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

此操作的輸出會列出您刪除之叢集快照的詳細資料。

管理亞馬遜文檔數據庫資源

這些章節涵蓋了管理 Amazon DocumentDB (具有 MongoDB 相容性) 實作的各種元件及其相關任務。

主題

- [Amazon DocumentDB 營運任務概觀](#)
- [Amazon DocumentDB 全球集群概述](#)
- [管理 Amazon DocumentDB 集群](#)
- [管理 Amazon DocumentDB 實例](#)
- [管理 Amazon DocumentDB 子網路群組](#)
- [Amazon DocumentDB 高可用性和複寫](#)
- [管理 Amazon DocumentDB 索引](#)
- [管理集合層級文件壓縮](#)
- [Amazon DocumentDB 事件範例事件管理](#)
- [選擇區域與可用區域](#)
- [管理 Amazon DocumentDB 叢集參數群組](#)
- [了解亞馬遜文檔數據庫端點](#)
- [了解 Amazon DocumentDB Amazon 資源名稱 \(ARN \)](#)
- [標記亞馬遜文檔數據庫資源](#)
- [維護 Amazon DocumentDB](#)
- [了解服務連結角色](#)

Amazon DocumentDB 營運任務概觀

本節涵蓋 Amazon DocumentDB (具有 MongoDB 相容性) 叢集的營運任務，以及如何使用 AWS CLI。

主題

- [新增複本到 Amazon DocumentDB 叢集](#)
- [管理叢集和執行個體](#)
- [建立叢集快照](#)
- [從快照還原](#)

- [從叢集中移除執行個體](#)
- [刪除叢集](#)

新增複本到 Amazon DocumentDB 叢集

您為 Amazon DocumentDB 叢集建立主要執行個體後，就可以新增一個或多個副本。複本是唯讀的執行個體，有兩種用途：

- 可擴展性— 如果您有大量的用戶端需要並行存取，您可以為讀取擴展新增更多複本。
- 高可用性— 如果主要執行個體失敗，Amazon DocumentDB 會自動容錯移轉到複本執行個體，並將其指定為新的主要執行個體。若複本失敗，叢集中的其他執行個體仍然可以處理請求，直到可復原失敗的節點為止。

每個 Amazon DocumentDB 叢集可支援達 15 個複本。

Note

為取得最大的容錯能力，建議您將複本部署在分離的可用區域中。這有助於確保即使整個可用區域都不可使用時，Amazon DocumentDB 叢集仍然能夠繼續運作。

以下 AWS CLI 範例說明如何新增新複本。--availability-zone 參數會取代所指定可用區域中的複本。

```
aws docdb create-db-instance \  
  --db-instance-identifier sample-instance \  
  --db-cluster-identifier sample-cluster \  
  --engine docdb \  
  --db-instance-class db.r5.large \  
  --availability-zone us-east-1a
```

管理叢集和執行個體

如下所示 AWS CLI 範例列出了一個區域中的所有 Amazon DocumentDB 叢集。對於某些管理功能 (如叢集和執行個體生命週期管理)，Amazon DocumentDB 利用與 Amazon RDS 共用的操作技術。所以此 filterName=engine, Values=docdb 篩選參數僅傳回 Amazon DocumentDB 叢集。

如需描述和修改叢集的詳細資訊，請參閱[Amazon DocumentDB 集群生命週](#)。

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

此操作的輸出將會如下所示。

```
{
  "DBClusters": [
    {
      "AvailabilityZones": [
        "us-east-1c",
        "us-east-1b",
        "us-east-1a"
      ],
      "BackupRetentionPeriod": 1,
      "DBClusterIdentifier": "sample-cluster-1",
      "DBClusterParameterGroup": "sample-parameter-group",
      "DBSubnetGroup": "default",
      "Status": "available",
      ...
    },
    {
      "AvailabilityZones": [
        "us-east-1c",
        "us-east-1b",
        "us-east-1a"
      ],
      "BackupRetentionPeriod": 1,
      "DBClusterIdentifier": "sample-cluster-2",
      "DBClusterParameterGroup": "sample-parameter-group",
      "DBSubnetGroup": "default",
      "Status": "available",
      ...
    },
    {
      "AvailabilityZones": [
        "us-east-1c",
        "us-east-1b",
        "us-east-1a"
      ],
      "BackupRetentionPeriod": 1,
      "DBClusterIdentifier": "sample-cluster-3",
      "DBClusterParameterGroup": "sample-parameter-group",

```

```
        "DBSubnetGroup": "default",
        "Status": "available",
        ...
    }
]
}
```

如下所示AWS CLI範例列出了 Amazon DocumentDB 叢集中的執行個體。如需描述和修改叢集的詳細資訊，請參閱[Amazon DocumentDB 實例生命週](#)。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[DBClusterMembers]'
```

輸出類似如下所示。這個輸出中有兩個執行個體。主要執行個體為 sample-instance-1 ("IsClusterWriter": true)。另外還有複本執行個體 sample-instance2 ("IsClusterWriter": false)。

```
[
  [
    [
      {
        "DBInstanceIdentifier": "sample-instance-1",
        "IsClusterWriter": true,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      },
      {
        "DBInstanceIdentifier": "sample-cluster-2",
        "IsClusterWriter": false,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      }
    ]
  ]
]
```


建立叢集快照

一個叢集快照是您 Amazon DocumentDB 叢集中的完整資料備份。當快照建立時，Amazon DocumentDB 會直接從叢集磁盤區讀取您的資料。因此，即使叢集當時並未執行任何執行個體，您仍可以建立快照。建立快照所需的時間量取決於叢集磁碟區的大小。

Amazon DocumentDB 支援自動備份，這會在每天偏好的備份時段發生每天一段 30 分鐘的時間。以下 AWS CLI 範例說明如何檢視叢集的備份時段：

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].PreferredBackupWindow'
```

輸出會顯示備份時段 (UTC)：

```
[  
  "00:18-00:48"  
]
```

您可以在建立 Amazon DocumentDB 叢集時定義備份時段。您也可以變更備份時段，如下列範例所示。如果您未定義備份時段，Amazon DocumentDB 會自動為您的叢集指派備份時段。

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --preferred-backup-window "02:00-02:30"
```

除了自動備份之外，您可以隨時手動建立叢集快照。當您執行此操作時，您需要指定要備份的叢集，以及快照的唯一名稱，以便之後能夠還原。

以下 AWS CLI 範例說明如何建立資料的快照。

```
aws docdb create-db-cluster-snapshot \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

從快照還原

您可以將叢集快照還原到新的 Amazon DocumentDB 叢集。若要執行這項操作，請提供快照集的名稱和新叢集的名稱。您無法從快照還原到現有叢集；而是會在您還原時建立新叢集，然後使用快照資料填入它。

以下範例顯示叢集 `sample-cluster` 的所有快照，

```
aws docdb describe-db-cluster-snapshots \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusterSnapshots[*].[DBClusterSnapshotIdentifier,SnapshotType,Status]'
```

輸出類似如下所示。手動快照是您手動建立的快照，而自動快照由 Amazon DocumentDB 在叢集備份時段內建立。

```
[  
  "sample-cluster-snapshot",  
  "manual",  
  "available"  
],  
 [  
  "rds:sample-cluster",  
  "automated",  
  "available"  
 ]  
]
```

下列範例示範如何從快照還原 Amazon DocumentDB 叢集。

```
aws docdb restore-db-cluster-from-snapshot \  
  --engine docdb \  
  --db-cluster-identifier new-sample-cluster \  
  --snapshot-identifier sample-cluster-snapshot
```

新叢集沒有任何相關聯的執行個體；因此，如果您想要與叢集互動，必須在當中新增執行個體。

```
aws docdb create-db-instance \  
  --db-instance-identifier new-sample-instance \  
  --db-instance-class db.r5.large \  
  --engine docdb \  
  --db-cluster-identifier new-sample-cluster
```

您可以使用以下 AWS CLI 操作來監控叢集和執行個體建立的進度。當叢集和執行個體狀態都可取得時，您就可以連接至新叢集的端點並存取您的資料。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier new-sample-cluster \  
  --query 'DBClusters[*].[Status,Endpoint]'
```

```
aws docdb describe-db-instances \  
  --db-instance-identifier new-sample-instance \  
  --query 'DBInstances[*].[DBInstanceStatus]'
```

從叢集中移除執行個體

Amazon DocumentDB 會將您的所有資料存放在叢集磁碟區中。資料會保留在該叢集磁碟區中，即使您移除叢集中的所有執行個體也一樣。如果您需要再次存取資料，可以隨時新增執行個體至叢集，並從您上次離開的位置繼續操作。

下列範例示範如何從您的 Amazon DocumentDB 叢集中移除執行個體。

```
aws docdb delete-db-instance \  
  --db-instance-identifier sample-instance
```

刪除叢集

您必須先移除所有執行個體，才能刪除 Amazon DocumentDB 叢集。以下 AWS CLI 範例會傳回有關叢集中執行個體的資訊。如果此操作傳回任何執行個體識別碼，則您必須刪除每個執行個體。如需詳細資訊，請參閱 [從叢集中移除執行個體](#)。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].DBClusterMembers[*].DBInstanceIdentifier'
```

當沒有剩餘的執行個體時，您就可以刪除叢集。同時，您必須選擇以下其中一個選項：

- 建立最終快照— 捕獲快照中的所有叢集資料，如此您稍後就可以利用該資料重新建立新的執行個體。下列範例示範其做法：

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster
```

```
--db-cluster-identifier sample-cluster \  
--final-db-snapshot-identifier sample-cluster-snapshot
```

- 跳過最終快照— 永久捨棄所有叢集資料。這個動作無法還原。下列範例示範其做法：

```
aws docdb delete-db-cluster \  
--db-cluster-identifier sample-cluster \  
--skip-final-snapshot
```

Amazon DocumentDB 全球集群概述

什麼是全域叢集？

全域叢集由一個主要區域和最多五個唯讀次要區域組成。您可以直接向主要區域中的主要叢集發出寫入操作，Amazon DocumentDB 會使用專用基礎設施自動將資料複寫到次要區域。延遲通常不到一秒鐘。

全局集群如何有用？

- 從全區域中斷的復原 — 如果發生全區域中斷，您可以在幾分鐘內將其中一個次要叢集提升至主要叢集，典型的復原時間目標 (RTO) 不到一分鐘。復原點目標 (RPO) 通常以秒為單位測量，但這取決於發生故障時網路上的延遲情況。
- 具有本機延遲的全域讀取 — 如果您在世界各地設有辦公室，則可以使用全域叢集來保持主要區域中的主要資訊來源更新。您其他地區的辦公室可以存取自己所在地區的資訊，並具有本機延遲時間。
- 可擴充的次要叢集 — 您可以將更多唯讀執行個體新增至次要區域來擴展次要叢集。次要叢集是唯讀的，因此最多可支援 16 個唯讀複本執行個體，而不是單一叢集的通常限制為 15。
- 從主要叢集到次要叢集的快速複寫 — 全域叢集執行的複寫對主要資料庫叢集的效能影響很小。資料庫執行個體的資源完全投入處理應用程式讀取與寫入工作負載。

全域叢集目前有哪些限制？

- Amazon DocumentDB 3.6 版不支援全域叢集。
- t3、t4g 和 r4 執行個體類型不支援全域叢集。
- 以下地區不提供全球叢集：南美洲 (聖保羅)、歐洲 (米蘭)、中國 (北京) 和中國 (寧夏)。
- 如果發生區域容錯移轉，您必須手動將次要叢集升級為主要叢集，然後修改應用程式以指向新的主要叢集。

- 只有主要叢集會執行寫入操作。執行寫入作業的用戶端會連線至主要叢集的叢集端點。
- 您的叢集最多可以有五個次要區域和一個主要區域。
- 次要叢集無法停止。如果主要叢集具有相關聯的次要叢集，則無法停止該叢集。只能停止沒有次要叢集的區域叢集。
- 附加至次要叢集的複本可在特定情況下重新啟動。如果主要區域的執行個體重新啟動或容錯移轉，次要區域中的複本也會重新啟動。然後，在所有複本與主要資料庫叢集的寫入器執行個體恢復同步之前，叢集將無法使用。此行為在預期當中。在變更主要叢集之前，請務必瞭解對全域叢集的影響。
- 您無法在次要叢集上使用變更串流。

主題

- [快速入門指南：全域叢集](#)
- [管理亞 Amazon DocumentDB 全域叢集](#)
- [Connect 到 Amazon DocumentDB 全球集群](#)
- [監控 Amazon DocumentDB 全球集群](#)
- [災難復原和 Amazon DocumentDB 全球叢集](#)

快速入門指南：全域叢集

主題

- [組態](#)
- [建立 Amazon DocumentDB 全域叢集](#)
- [新增AWS 區域至 Amazon DocumentDB 全域叢集](#)
- [為您的 Amazon DocumentDB 全域叢集使用快照](#)

組態

Amazon DocumentDB 全球叢集至少涵蓋兩個。AWS 區域主要區域支援具有一個主要 (寫入器) 執行個體和最多十五個複本執行個體的叢集，而次要區域則執行完全由十六個複本執行個體組成的唯讀叢集。全域叢集最多可以有五個次要區域。此表格列出全域叢集中允許的叢集、執行處理和複本上限。

| 描述 | 主要 AWS 區域 | 次要 AWS 區域 |
|----|-----------|-----------|
| 叢集 | 1 | 5 (最大值) |

| 描述 | 主要 AWS 區域 | 次要 AWS 區域 |
|--|-----------|-------------------|
| 寫入器執行個體 | 1 | 0 |
| 每個叢集的唯一讀執行個體 (Amazon DocumentDB 複本) | 15 (上限) | 16 (總計) |
| 唯讀執行個體 (允許的上限, 給定的次要區域的實際數量) | 15 - s | s = 次要 AWS 區域 的總數 |

叢集具有下列特定需求：

- 資料庫執行個體類別需求 — 您只能使用db.r5和db.r6執行個體類別。
- AWS 區域需求 — 主要叢集必須位於一個區域中，且至少一個次要叢集必須位於同一帳戶的不同區域中。您最多可以建立五個次要 (唯讀) 叢集，而且每個叢集都必須位於不同的區域。換句話說，沒有兩個叢集可以位於同一個區域。
- 命名需求 — 您為每個叢集選擇的名稱在所有區域中都必須是唯一的。您不能為不同的叢集使用相同的名稱，即使它們位於不同的區域。

建立 Amazon DocumentDB 全域叢集

您準備好建立第一個全域叢集了嗎？在本節中，我們將說明如何使用AWS Management Console或AWS CLI與下列指示建立具有新資料庫叢集和執行個體的全新全域叢集。

使用 AWS Management Console

1. 在中AWS Management Console，導覽至 Amazon DocumentDB。
2. 當您進入 Amazon DocumentDB 主控台時，請選擇 [叢集]。

The screenshot shows the AWS Management Console interface for Amazon DocumentDB. On the left, the navigation sidebar has 'Clusters' highlighted with a red circle. The main content area displays 'DocumentDB > Clusters' and 'Clusters (11)'. Below this is a search bar for 'Filter Resources' and a table of clusters. The 'Create' button in the top right corner of the cluster list is also circled in red.

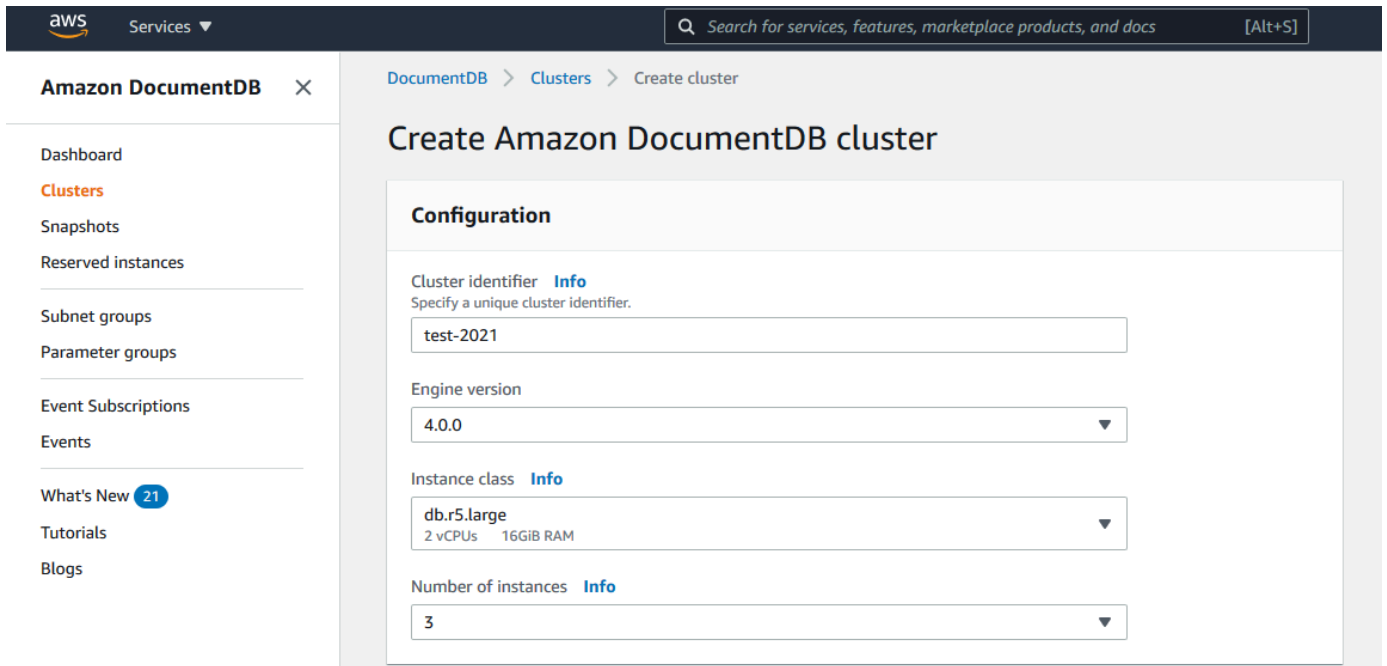
3. 選擇建立。

The screenshot shows the 'Create' button in the top right corner of the cluster list, circled in red. Below the button is a table with columns: Role, Engine version, Region & AZ, Status, Size, and Maintenance.

| Role | Engine version | Region & AZ | Status | Size | Maintenance |
|------------------|----------------|-------------|-----------|------------|-------------|
| Global cluster | 4.0.0 | 3 regions | available | 3 clusters | - |
| Regional cluster | 4.0.0 | us-east-2 | available | 1 Instance | None |
| Global cluster | 4.0.0 | 3 regions | available | 3 clusters | - |

4. 相應地填寫建立 Amazon DocumentDB 叢集表單的組態區段：

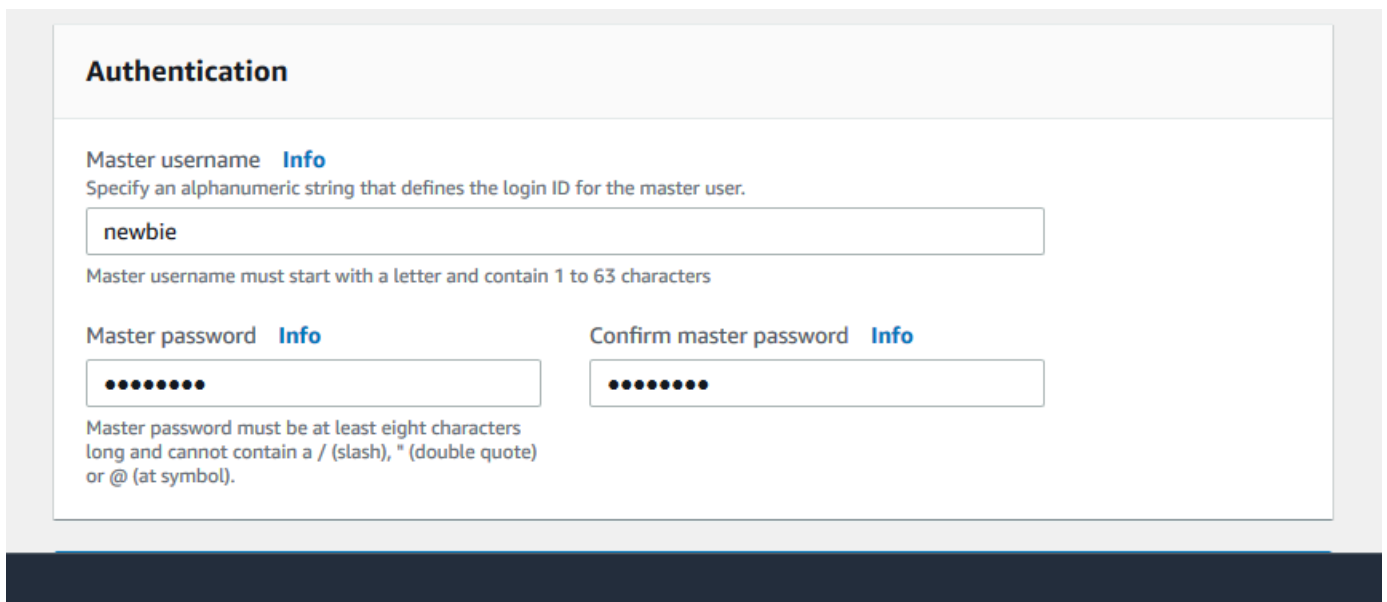
- 叢集識別碼：您可以為此執行個體輸入唯一識別碼，或允許 Amazon DocumentDB 根據叢集識別碼提供執行個體識別碼。
- 引擎版本：選擇 4.0.0
- 執行個體類別：選擇資料庫 .r5. 大
- 執行個體數目：選擇 3。



The screenshot shows the AWS Management Console interface for creating a new Amazon DocumentDB cluster. The breadcrumb navigation indicates the path: DocumentDB > Clusters > Create cluster. The main heading is "Create Amazon DocumentDB cluster". The "Configuration" section contains the following fields:

- Cluster identifier** (Info): Specify a unique cluster identifier. The input field contains "test-2021".
- Engine version**: A dropdown menu showing "4.0.0".
- Instance class** (Info): A dropdown menu showing "db.r5.large" with subtext "2 vCPUs 16GiB RAM".
- Number of instances** (Info): A dropdown menu showing "3".

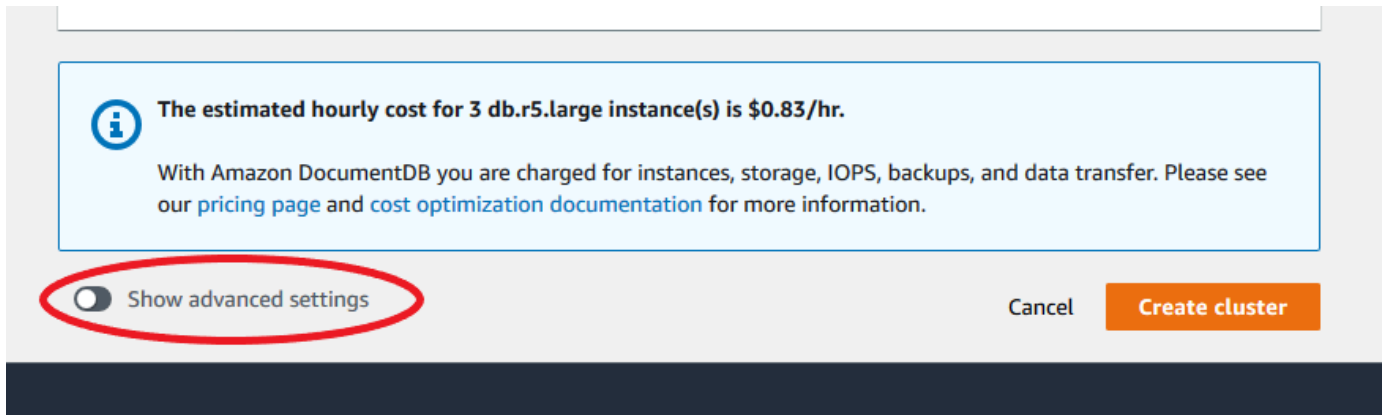
5. 在「驗證」區段中，填入主要使用者名稱和主密碼。



The screenshot shows the "Authentication" section of the AWS Management Console. It contains the following fields and instructions:

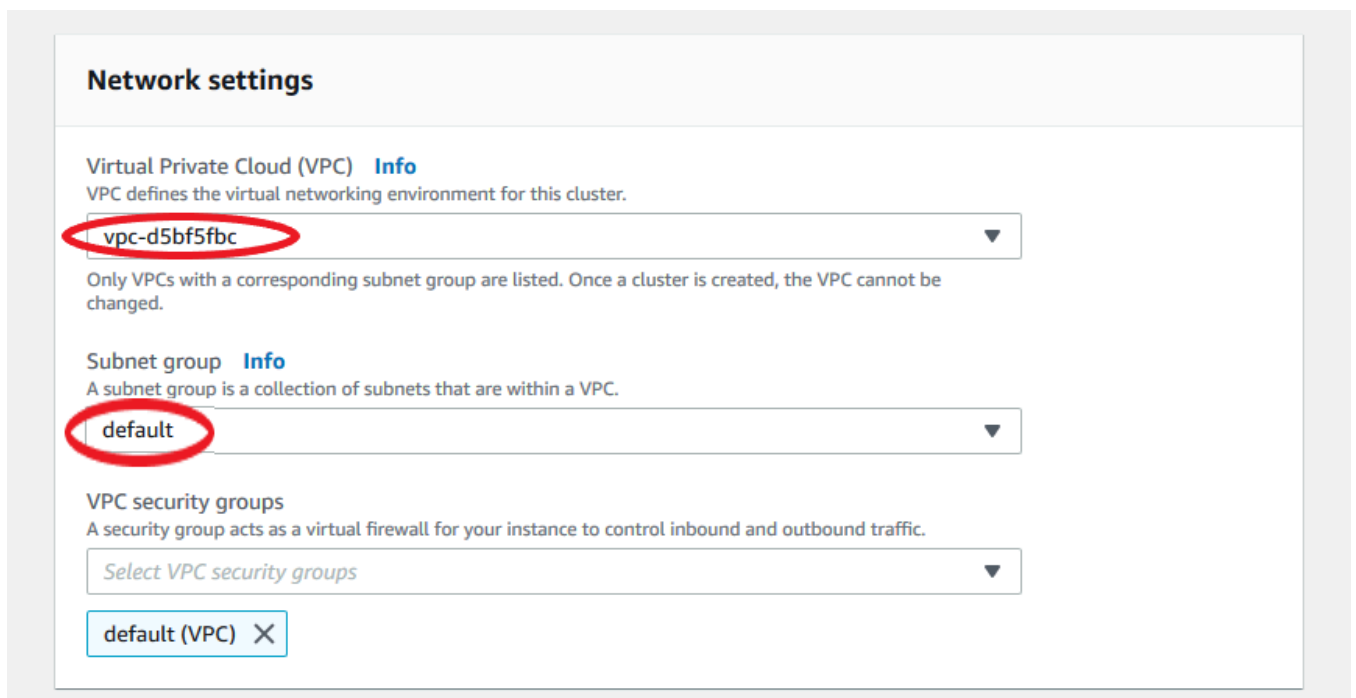
- Master username** (Info): Specify an alphanumeric string that defines the login ID for the master user. The input field contains "newbie". Below the field, it states: "Master username must start with a letter and contain 1 to 63 characters".
- Master password** (Info): An input field with masked characters (dots).
- Confirm master password** (Info): An input field with masked characters (dots).
- Below the password fields, it states: "Master password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol)." (Note: the original image has a typo in the quote character).

6. 選擇「顯示進階設定」。

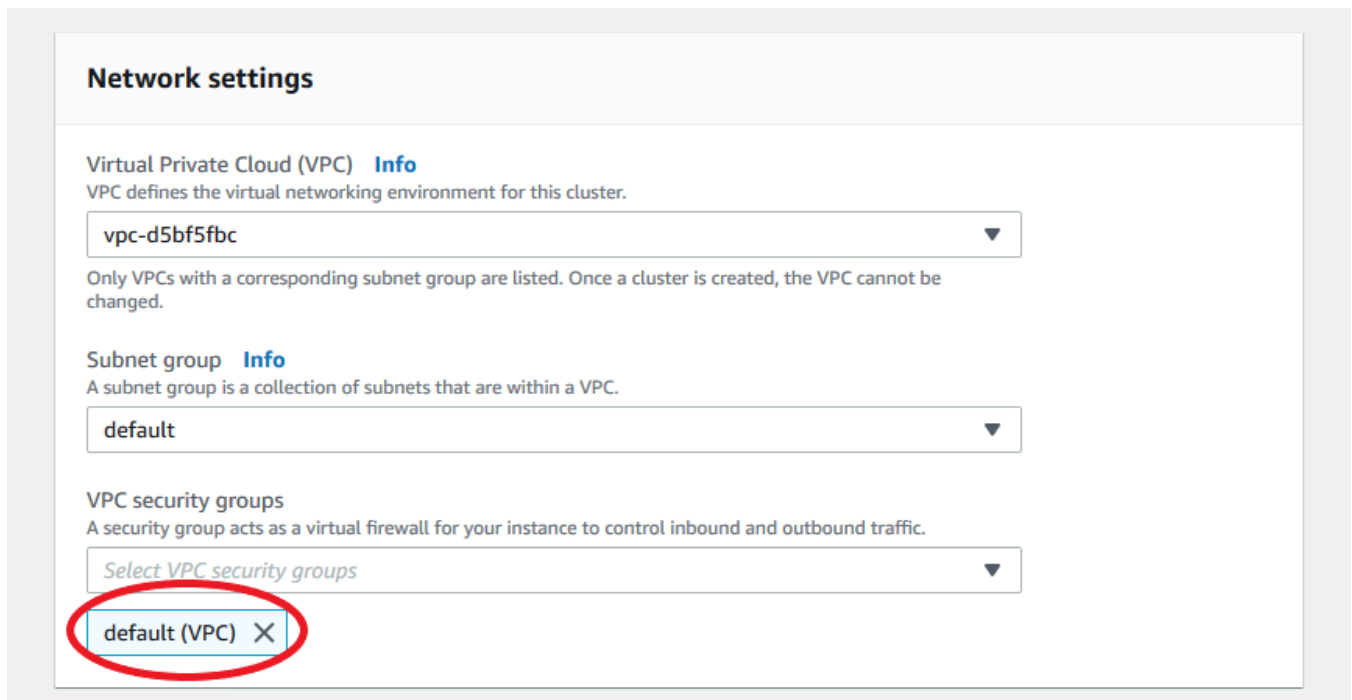


7. 在「網路設定」區段中：

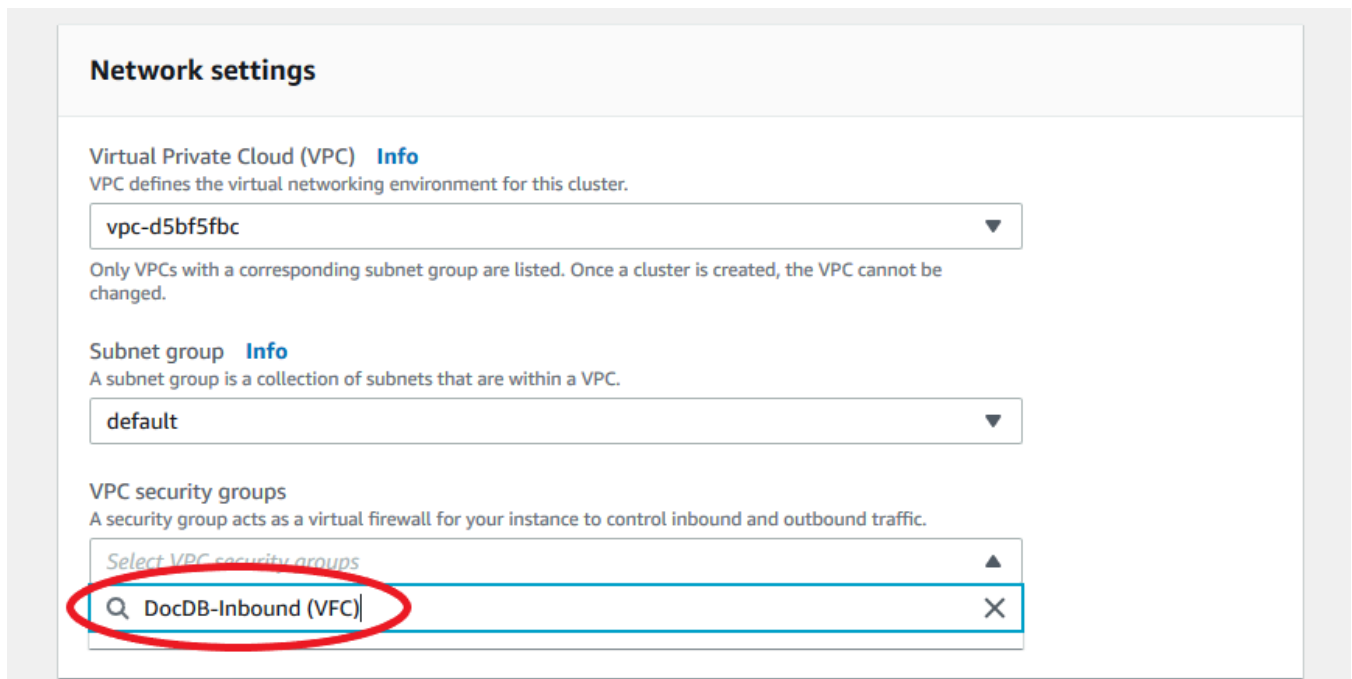
- 保留虛擬私有雲和子網路群組的預設選項。



- 對於 VPC 安全群組，應該已新增預設 VPC。



- 在「VPC 安全群組」欄位DocDB中輸入，然後選取「文件資料庫入埠 (VPC)」。



8. 對於「叢集選項」和「Encryption-at-rest」，請保留預設選項。

Cluster options

Port
TCP/IP port that is used to connect to the cluster.

27017

Cluster parameter group [Info](#)

default.docdb4.0

Encryption-at-rest

Encryption-at-rest [Info](#)

Enable encryption

Disable encryption

Master key

(default) aws/rds

Account

827630067164

KMS key ID

5e5dbe6b-e29d-4cfd-bfe5-585582908728

9. 對於「Backup」和「記錄匯出」，請保留預設選項。

Backup

Backup retention period [Info](#)

A period between 1 and 35 days in which you can perform a point-in-time restore and for which automated backups are retained.

1 day ▼

Backup window

The daily time range (in UTC) during which automated backups are created.

Start time

00 ▼

:

00 ▼

UTC

Duration

0.5 ▼

hours

Log exports

Select the log types to publish to Amazon CloudWatch Logs

Audit logs


Profiler logs

IAM role

The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS Service Linked Role

i To enable auditing, ensure that both exporting auditing logs to Amazon CloudWatch is enabled and the Cluster Parameter "Auditing" is enabled.

[Learn more](#) 

10. 對於維護、標籤和刪除保護，請保留預設選項。

Maintenance

Maintenance window [Info](#)
The period in which pending modifications or patches are applied to Instances in the cluster.

Select window

No preference

Tags

No tags

[Add tag](#)

Deletion protection

Enable deletion protection
Protects the cluster from being accidentally deleted. While this option is enabled, you can't delete the cluster.

11. 現在，單擊顯示「創建」的按鈕。

i The estimated hourly cost for 3 db.r5.large instance(s) is \$0.83/hr.

With Amazon DocumentDB you are charged for instances, storage, IOPS, backups, and data transfer. Please see our [pricing page](#) and [cost optimization documentation](#) for more information.

Show advanced settings

Cancel [Create cluster](#)

使用 AWS CLI

若要建立 Amazon DocumentDB 區域叢集，請呼叫 `create-db-cluster` AWS CLI 下列 AWS CLI 命令會建立名為的 Amazon DocumentDB 叢集。 `global-cluster-id` 如需刪除保護的詳細資訊，請參閱 [刪除 Amazon DocumentDB 集群](#)。

此外，這`--engine-version`是一個可選參數，默認為最新的主要引擎版本。目前的主要引擎版本為4.0.0。當新的主要引擎版本發行時，的預設引擎版本`--engine-version`將會更新，以反映持續的主要引擎版本。因此，對於生產工作負載，尤其是依賴指令碼、自動化或AWS CloudFormation範本的工作負載，我們建議您明確指定`--engine-version`預期的主要版本。

如果未指定`db-subnet-group-name`或`vpc-security-group-id`，Amazon DocumentDB 將使用指定區域的預設子網路群組和 Amazon VPC 安全群組。

在下列範例中，將每個#####取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster \  
  --global-cluster-identifier global-cluster-id \  
  --source-db-cluster-identifier arn:aws:rds:us-east-1:111122223333:cluster-id
```

針對 Windows：

```
aws docdb create-db-cluster ^  
  --global-cluster-identifier global-cluster-id ^  
  --source-db-cluster-identifier arn:aws:rds:us-east-1:111122223333:cluster-id
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBCluster": {  
    "StorageEncrypted": false,  
    "DBClusterMembers": [],  
    "Engine": "docdb",  
    "DeletionProtection" : "enabled",  
    "ClusterCreateTime": "2018-11-26T17:15:19.885Z",  
    "DBSubnetGroup": "default",  
    "EngineVersion": "4.0.0",  
    "MasterUsername": "masteruser",  
    "BackupRetentionPeriod": 1,  
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:cluster-id",  
    "DBClusterIdentifier": "cluster-id",  
    "MultiAZ": false,  
    "DBClusterParameterGroup": "default.docdb4.0",
```

```
    "PreferredBackupWindow": "09:12-09:42",
    "DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVLNTU7XVAY",
    "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
    "Port": 27017,
    "Status": "creating",
    "ReaderEndpoint": "cluster-id.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
    "AssociatedRoles": [],
    "HostedZoneId": "ZNKXTT8WH85VW",
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-77186e0d",
        "Status": "active"
      }
    ],
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1c",
      "us-east-1e"
    ],
    "Endpoint": "cluster-id.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
  }
}
```

建立叢集需要幾分鐘才能完成。您可以使用AWS Management Console或 AWS CLI 監控叢集的狀態。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 集群的狀態](#)。

Important

當您使用建立 Amazon DocumentDB 區域叢集時，不會建立任何執行個體。AWS CLI 因此，您必須明確建立主要執行個體，以及您需要的任何複本執行個體。您可以使用主控台或 AWS CLI 建立執行個體。如需詳細資訊，請參閱 Amazon DocumentDB API 參考 [CreateDBCluster](#) 中的 [將 Amazon DocumentDB 執行個體新增至叢集](#)。

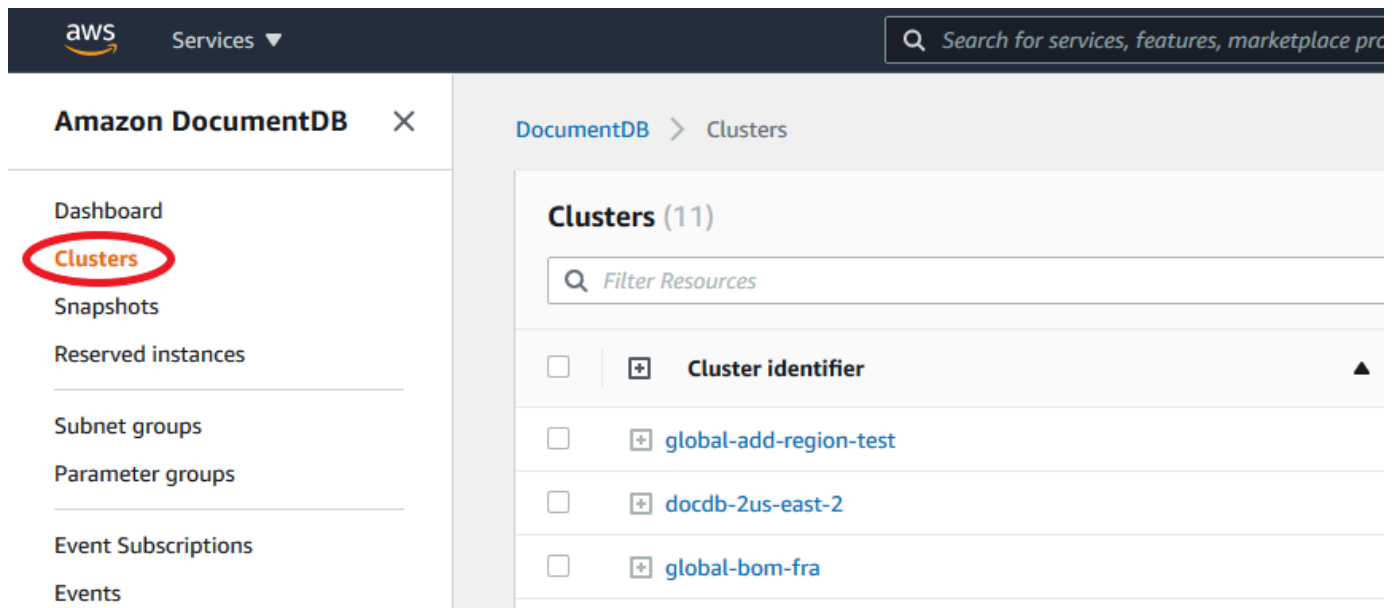
當您的地區叢集可用時，您可以依照下列指示在其他區域新增次要叢集：[新增AWS 區域至 Amazon DocumentDB 全域叢集](#)。當您新增區域時，您的地區叢集會成為您的主要叢集，而您在選擇的區域中有一個新的次要叢集。

新增AWS 區域至 Amazon DocumentDB 全域叢集

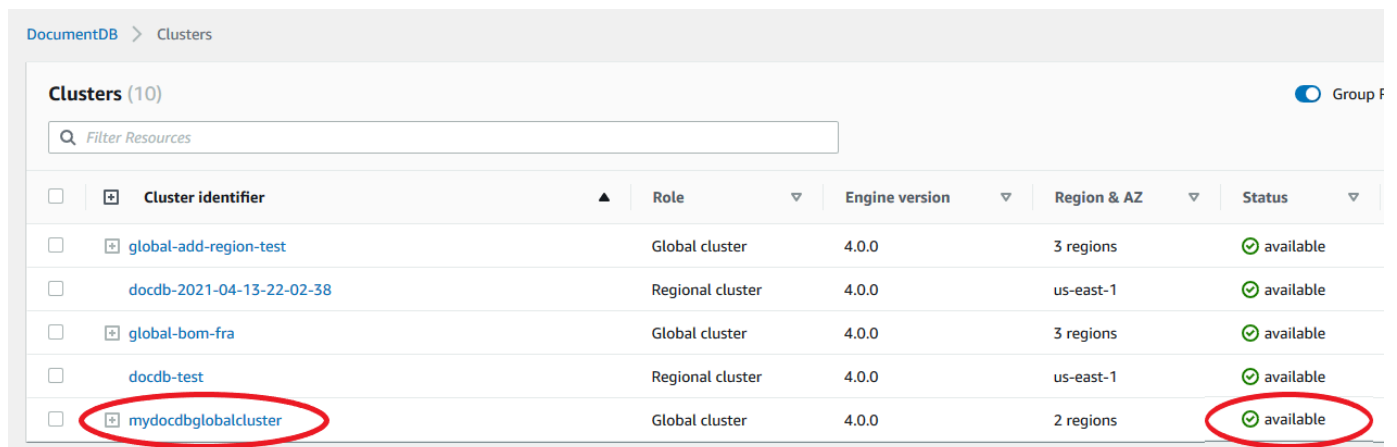
全域叢集在與主要叢集不同的區域中至少需要一個次要叢集，而且您最多可以新增五個次要叢集。請注意，對於您新增的每個次要叢集，您必須將主要叢集中允許的複本數目減少一個。例如，如果您的全域叢集有五個次要區域，則主要叢集只能有十個（而非十五個）複本。如需詳細資訊，請參閱 [Amazon DocumentDB 全域叢集的組態需求](#)。

使用 AWS Management Console

1. 登入AWS Management Console並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。



3. 選擇您要新增次要叢集的叢集。請確定叢集為Available。



4. 選取「作業」的下拉式功能表，然後選擇「新增區域」。

DocumentDB > Clusters

Clusters (10) Group Resources Actions ▲ Create

| <input type="checkbox"/> | Cluster identifier ▲ | Role ▼ | Engine version ▼ | Region & AZ ▼ | Status ▼ | Add Region | Maintenance ▼ |
|-------------------------------------|---------------------------|------------------|------------------|---------------|-----------|--|---------------|
| <input type="checkbox"/> | global-add-region-test | Global cluster | 4.0.0 | 3 regions | available | 3 clusters | - |
| <input type="checkbox"/> | docdb-2021-04-13-22-02-38 | Regional cluster | 4.0.0 | us-east-1 | available | 0 Instances | None |
| <input type="checkbox"/> | global-bom-fra | Global cluster | 4.0.0 | 3 regions | available | 3 clusters | - |
| <input type="checkbox"/> | docdb-test | Regional cluster | 4.0.0 | us-east-1 | available | 0 Instances | None |
| <input checked="" type="checkbox"/> | mydocdbglobalcluster | Global cluster | 4.0.0 | 2 regions | available | 2 clusters | - |

- 在「新增區域」頁面上，選擇次要區域。請注意，您無法為同一個全域叢集選擇已有次要叢集的區域。此外，它不能與主要叢集位於同一個區域。如果這是您要新增的第一個區域，您也必須指定您選擇的全域叢集識別碼。

DocumentDB > Clusters > Add region

Add an AWS Region

You are adding a secondary region to your Amazon DocumentDB global cluster. Secondary Regions can serve low latency reads. In the unlikely event that your database becomes degraded or isolated in the primary region, you can promote your secondary region

AWS Region

Secondary region

Select one - ▼

- 完成新區域中次要叢集的其餘欄位，然後選取建立叢集。完成新增區域之後，您可以在中的叢集清單中看到它AWS Management Console。

Configuration


Global Cluster Id
firstregion

Cluster identifier [Info](#)
Specify a unique cluster identifier.

Instance class [Info](#)

2 vCPUs 16GiB RAM

Number of instances [Info](#)

 **The estimated hourly cost for 3 db.r5.large instance(s) is \$0.83/hr.**

With Amazon DocumentDB you are charged for instances, storage, IOPS, backups, and data transfer. Please see our [pricing page](#) and [cost optimization documentation](#) for more information.

Show advanced settings

Cancel **Create cluster**

使用 AWS CLI

- 使用 `create-db-cluster` CLI 命令搭配全域叢集(`--global-cluster-identifier`)的名稱。對於其他參數，請執行下列動作：
 - 對於 `--region`，請選擇與您主要區域AWS 區域不同的區域。
 - 請選擇 `--engine` 和 `--engine-version` 參數的特定值。
 - 對於加密叢集，請指定主要 AWS 區域 作為要加密的 `--source-region`。

下列範例會建立新的 Amazon DocumentDB 叢集，並將其做為唯讀次要叢集附加到全域叢集。在最後一個步驟中，執行個體會新增至新叢集。

在下列範例中，將每個#####取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb --region secondary-region-id \  
  create-db-cluster \  
    --db-cluster-identifier cluster-id \  
    --global-cluster-identifier global-cluster-id \  
    --engine-version version \  
    --engine docdb  
  
aws docdb --region secondary-region-id \  
  create-db-instance \  
    --db-cluster-identifier cluster-id \  
    --global-cluster-identifier global-cluster-id \  
    --engine-version version \  
    --engine docdb
```

針對 Windows :

```
aws docdb --region secondary-region-id ^  
  create-db-cluster ^  
    --db-cluster-identifier cluster-id ^  
    --global-cluster-identifier global-cluster-id ^  
    --engine-version version ^  
    --engine docdb  
  
aws docdb --region secondary-region-id ^  
  create-db-instance ^  
    --db-cluster-identifier cluster-id ^  
    --global-cluster-identifier global-cluster-id ^  
    --engine-version version ^  
    --engine docdb
```

為您的 Amazon DocumentDB 全域叢集使用快照

您可以還原 Amazon DocumentDB 叢集的快照，以做為全域叢集的起點使用。若要這麼做，您必須還原快照並建立新叢集。這將作為全域叢集的主要叢集。然後，您可以將另一個區域新增至還原的叢集，從而將其轉換為全域叢集。

管理亞 Amazon DocumentDB 全域叢集

您可以在組成全域叢集的個別叢集上執行大部分的管理作業。當您在主控台的 [叢集] 頁面上選擇 [群組相關資源] 時，您會看到主要叢集和次要叢集在關聯的全域叢集下分組。

全域叢集的 [組態] 索引標籤會AWS 區域顯示叢集的執行位置、版本以及全域叢集識別碼。

主題

- [修改 Amazon DocumentDB 全域叢集](#)
- [修改 Amazon DocumentDB 全域叢集的參數](#)
- [從 Amazon DocumentDB 全域叢集中移除叢集](#)
- [從 Amazon DocumentDB 全域叢集中刪除叢集](#)
- [在次要區域中建立無周邊的 Amazon DocumentDB 叢集](#)

修改 Amazon DocumentDB 全域叢集

中的 [叢集] 頁面會AWS Management Console列出所有全域叢集，並顯示每個叢集的主要叢集和次要叢集。全域叢集有自己的組態設定。具體而言，它具有與其主要和次要叢集相關聯的區域。

當您對全域叢集進行變更時，您有機會取消變更。

當您選擇 Continue (繼續) 時，您可以確認變更。

修改 Amazon DocumentDB 全域叢集的參數

您可以針對全域叢集中的每個叢集個別設定叢集參數群組。大多數參數的運作方式與其他類型的 Amazon DocumentDB 叢集的運作方式相同。建議您將全域資料庫中所有叢集的設定保持一致。如果您將次要叢集提升為主要叢集，這麼做有助於避免非預期的行為變更。

例如，讓時區和字元集使用相同設定，以避免不同叢集接管成為主要叢集時發生不一致的行為。

從 Amazon DocumentDB 全域叢集中移除叢集

有幾種情況下，您可能會想要從全域叢集中移除叢集。例如，如果主要叢集降級或隔離，您可能想要從全域叢集移除叢集。然後，它會變成獨立佈建的叢集，可用來建立新的全域叢集。若要深入了解，請參閱從意外中斷手動復原全域叢集。

您也可能想要移除叢集，因為您想要刪除不再需要的全域叢集。在卸離所有關聯的叢集之後，才能刪除全域叢集，將主叢集保留為最後。如需詳細資訊，請參閱刪除 Amazon DocumentDB 全域叢集。

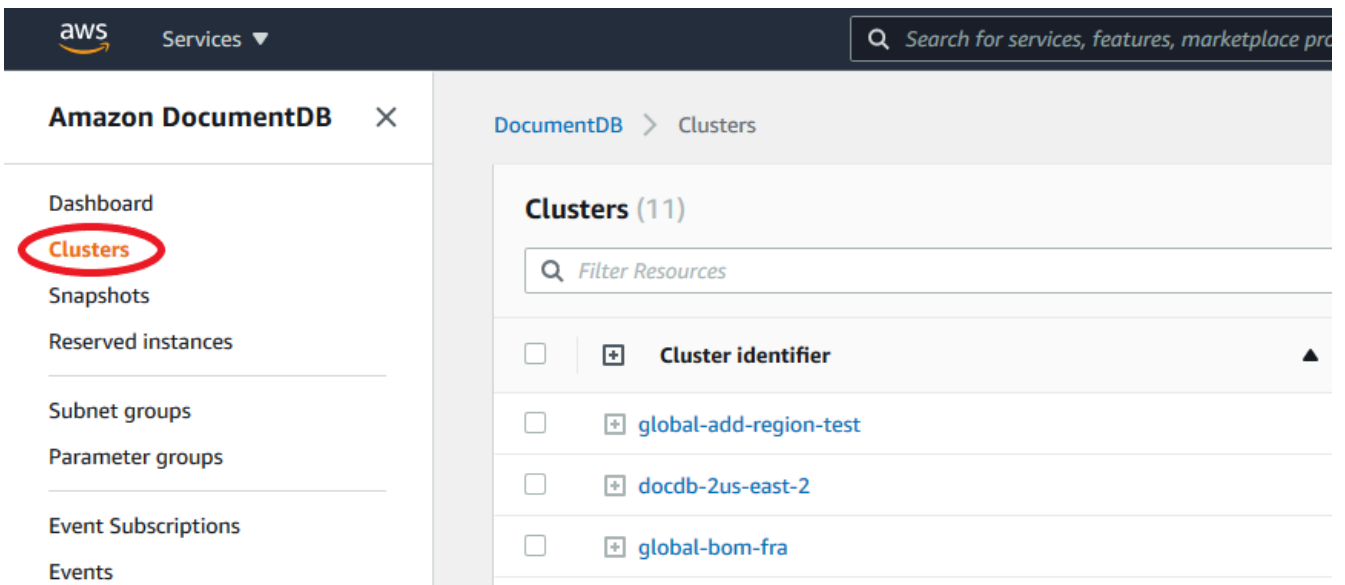
Note

當叢集從全域叢集中分離時，叢集不再與主叢集同步。它會成為具有完整讀取/寫入功能的獨立佈建叢集。此外，它不再顯示在 Amazon DocumentDB 主控台中。只有當您在叢集所在的主控台中選取區域時，此選項才會顯示。

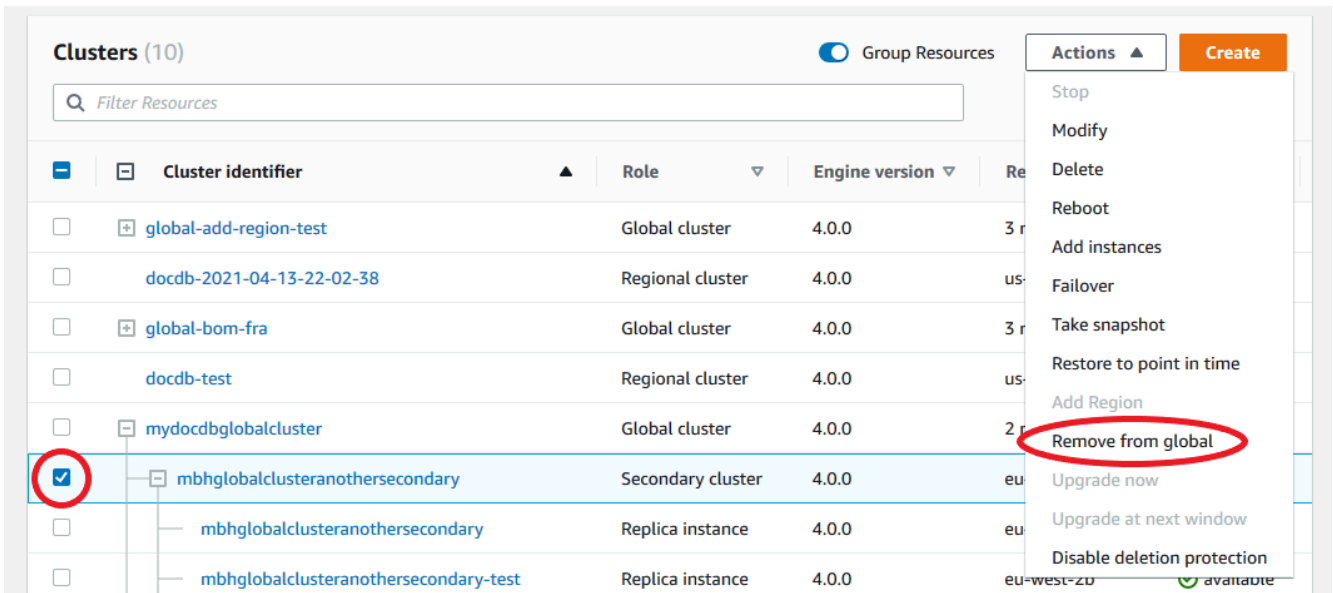
您可以使用AWS Management Console、或 RDS API 從全域叢集中AWS CLI移除叢集。

Using the AWS Management Console

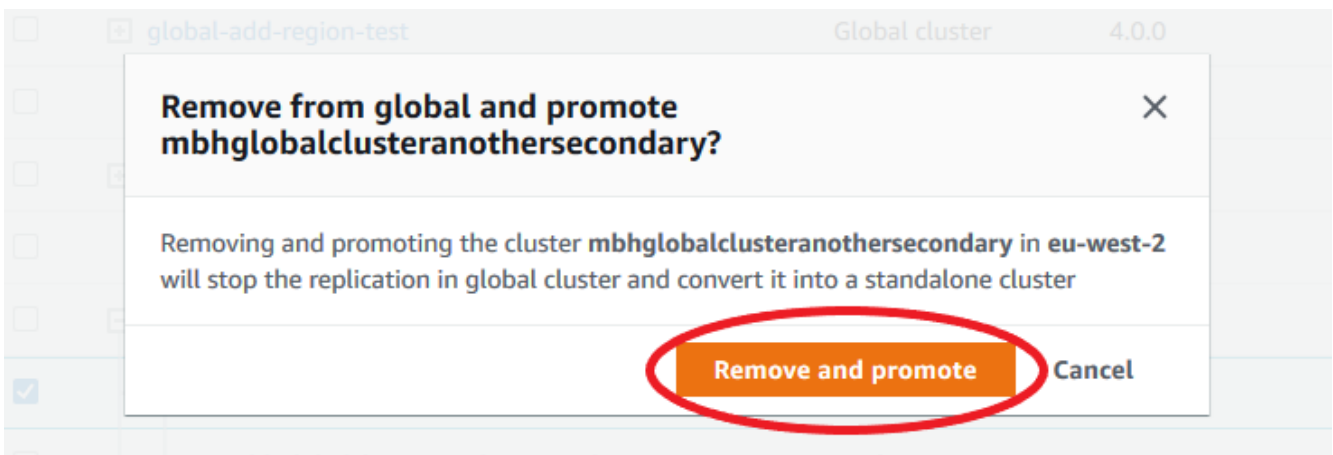
1. 登入AWS Management Console並瀏覽至 Amazon DocumentDB 主控台。
2. 在左側導覽中選擇 [叢集]。



3. 展開全域叢集，以便查看所有次要叢集。選取要移除的次要叢集。選擇「操作」，然後在下拉菜單中選擇「從全局中刪除」。



- 將會出現提示，要求您確認是否要將次要從全域叢集中斷連結。選擇移除並升級，從全域叢集移除叢集。



現在，該叢集不再作為次要叢集使用，而且不再與主要叢集同步。它是具有完全讀/寫功能的獨立集群。

移除或刪除所有次要叢集之後，您就可以用同樣的方式移除主要叢集。在移除所有次要叢集之後，您無法從全域叢集中卸離或移除主要叢集。全域叢集可能會保留在 [叢集] 清單中，區域和 AZ 為零。如果您不想再使用此全域叢集，可以刪除。

Using the AWS CLI

若要從全域叢集移除叢集，請使用下列參數執行 `remove-from-global-cluster` CLI 命令：

- `--global-cluster-identifier`— 全域叢集的名稱 (識別碼)。

- `--db-cluster-identifier`— 要從全域叢集移除的每個叢集名稱。

下列範例會先移除次要叢集，然後從全域叢集移除主要叢集。

若為 Linux、macOS 或 Unix：

```
aws docdb --region secondary_region \  
  remove-from-global-cluster \  
    --db-cluster-identifier secondary_cluster_ARN \  
    --global-cluster-identifier global_cluster_id  
  
aws docdb --region primary_region \  
  remove-from-global-cluster \  
    --db-cluster-identifier primary_cluster_ARN \  
    --global-cluster-identifier global_cluster_id
```

針對全域叢集中的每個次要區域重複此 `remove-from-global-cluster--db-cluster-identifiersecondary_cluster_ARN` 指令。

針對 Windows：

```
aws docdb --region secondary_region ^  
  remove-from-global-cluster ^  
    --db-cluster-identifier secondary_cluster_ARN ^  
    --global-cluster-identifier global_cluster_id  
  
aws docdb --region primary_region ^  
  remove-from-global-cluster ^  
    --db-cluster-identifier primary_cluster_ARN ^  
    --global-cluster-identifier global_cluster_id
```

針對全域叢集中的每個次要區域重複此 `remove-from-global-cluster--db-cluster-identifiersecondary_cluster_ARN` 指令。

從 Amazon DocumentDB 全域叢集中刪除叢集

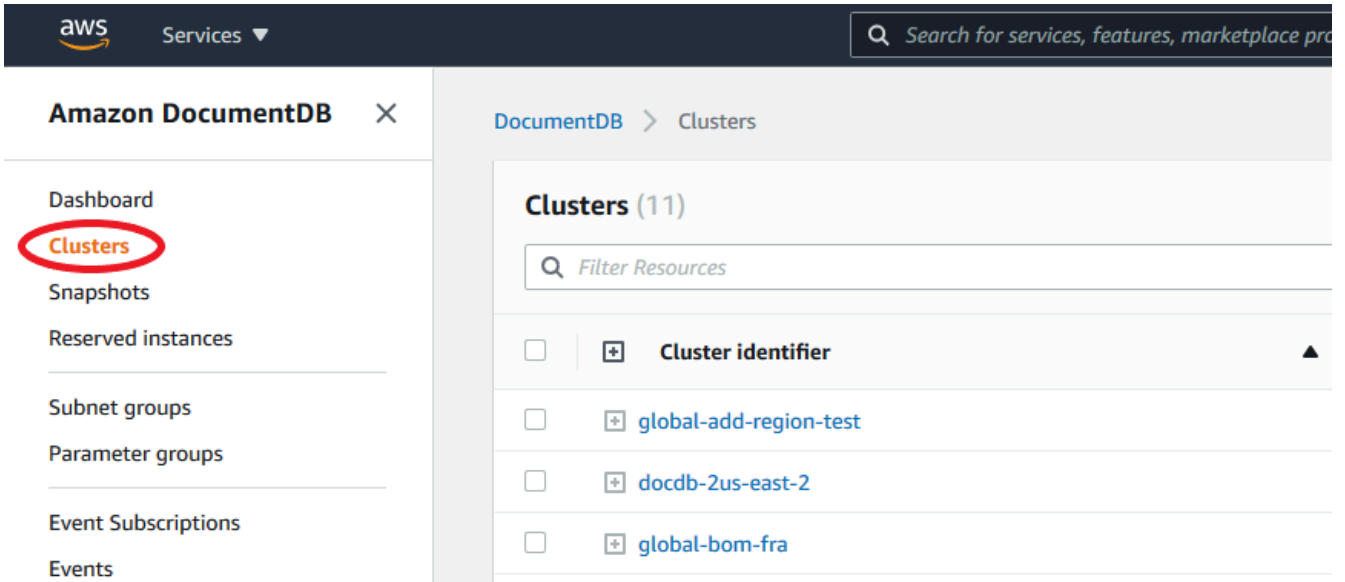
若要刪除全域叢集，請執行下列動作：

- 從全域叢集移除所有次要叢集。每個叢集都會成為獨立叢集。請參閱上一節「移除全域叢集」。
- 從每個獨立叢集中刪除所有複本。

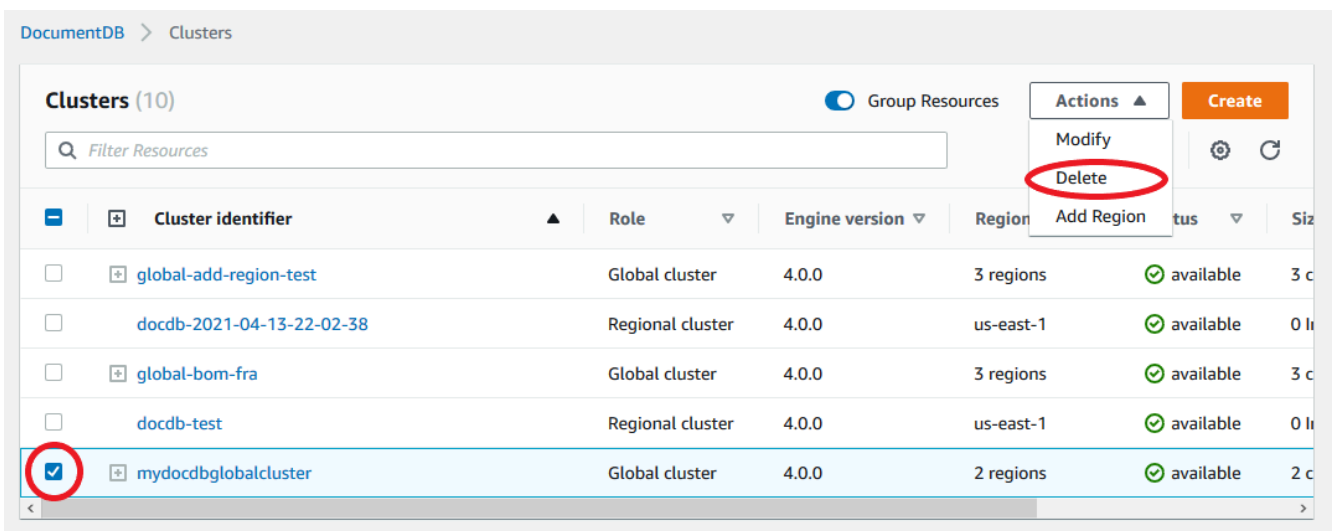
- 從全域叢集移除主要叢集。這會成為獨立叢集。
- 從主要叢集中，先刪除所有複本，然後刪除主要執行個體。從新獨立叢集中刪除主要執行個體通常也會同時移除叢集和全域叢集。

Using the AWS Management Console

1. 登入AWS Management Console並瀏覽至 Amazon DocumentDB 主控台。
2. 選擇 [叢集]，然後尋找要刪除的全域叢集。



3. 選取全域叢集後，從「動作」功能表選擇「刪除」。



確認已從全域叢集移除所有叢集。全域叢集應顯示零區域和 AZ，以及零叢集的大小。如果全域叢集包含任何叢集，您就無法刪除它。首先，您必須遵循上一步中的指示，移除全域叢集。

Using the AWS CLI

若要刪除全域叢集，請使用名稱AWS 區域和全域叢集識別碼執行 `delete-global-cluster` CLI 命令，如下列範例所示。

若為 Linux、macOS 或 Unix：

```
aws docdb --region primary_region delete-global-cluster \  
  --global-cluster-identifier global_cluster_id
```

針對 Windows：

```
aws docdb --region primary_region delete-global-cluster ^ \  
  --global-cluster-identifier global_cluster_id
```

在次要區域中建立無周邊的 Amazon DocumentDB 叢集

雖然 Amazon DocumentDB 全域叢集需要與主叢集不同AWS 區域的次要叢集至少有一個次要叢集，但是您可以為次要叢集使用無周邊設定。無周邊的次要 Amazon DocumentDB 叢集是沒有執行個體的叢集。這種類型的組態可以降低全域叢集的費用。在 Amazon DocumentDB 叢集中，運算和儲存是分離的。如果沒有執行個體，您就不需要支付運算費用，只需支付儲存體費用。如果設定正確，無周邊輔助磁碟區的儲存磁碟區會與主要叢集保持同步。

您可以像在建立 Amazon DocumentDB 全域叢集時一樣新增次要叢集。不過，在主要叢集開始複寫至次要叢集之後，您可以從次要叢集刪除唯讀執行個體。此次要叢集現在被視為「無頭」，因為它不再具有執行個體。然而，儲存磁碟區會與主要的 Amazon DocumentDB 叢集保持同步。

Important

對於能夠容忍全區域故障 15 分鐘以上的客戶，我們只建議使用無周邊叢集。這是因為使用無周邊次要叢集從全區域故障中復原時，使用者必須在容錯移轉後建立新的執行個體。新執行個體可能需要 10-15 分鐘才能使用。

如何將無周邊次要叢集新增至全域叢集

1. 登入AWS Management Console並開啟 [Amazon DocumentDB](#) 主控台。

2. 在左側導覽中選擇 [叢集]。
3. 選擇需要次要叢集的全域叢集。請確定主要叢集為Available。
4. 對於 動作 ，請選擇 Add region (新增區域)。
5. 在「新增區域」頁面上，選擇次要區域。

Note

您無法為同一個全域叢集選擇已有次要叢集的區域。此外，它不能與主要叢集位於同一個區域。

6. 完成新區域中次要叢集的剩餘欄位。這些選項與任何叢集執行個體的組態選項相同。
7. 新增區域。完成將區域新增至全域叢集之後，您會在中的Clusters清單中看到該區域AWS Management Console。
8. 在繼續之前，請使用AWS Management Console或檢查次要叢集及其讀取器執行個體的状态AWS CLI。如果您使用以下命令，則以下是一個示例命令AWS CLI：

```
$ aws docdb describe-db-clusters --db-cluster-identifier secondary-cluster-id --query '*[].[Status]' --output text
```

新增的次要叢集的状态可能需要幾分鐘的時間才會從建立變更為可用。當叢集可用時，您可以刪除讀取器執行個體。

9. 選取次要叢集中的讀取器執行個體，然後選擇 [刪除]。
10. 刪除讀取器執行個體之後，次要叢集仍然是全域叢集的一部分。它應該沒有與其相關聯的實例。

Note

您可以使用這個無周邊的次要 Amazon DocumentDB 叢集，在發生此類中斷時，從主要區域的意外中斷手動復原 Amazon DocumentDB 全域叢集。

Connect 到 Amazon DocumentDB 全球集群

連線到全域叢集的方式取決於您是否需要寫入叢集或從叢集讀取：

- 對於唯讀請求或查詢，您可以連接到AWS 區域。

- 若要執行資料操作語言 (DML) 或資料定義語言 (DDL) 陳述式，您需要連接到主要叢集的叢集端點。此端點與您的應用程式可能位於不同的 AWS 區域。

在主控台中檢視全域叢集時，您可以看到與其所有叢集相關聯的所有一般用途端點。

連線到全域叢集的方式取決於您是否需要寫入資料庫或從資料庫讀取。對於您想要從主要區域提供服務的 DDL、DML 和讀取作業，您應該連線到主要叢集。建議您以複本集模式使用叢集端點連線到主要叢集，讀取偏好設定為 `secondaryPreferred=true`。這會將寫入流量路由到主要叢集的寫入器執行個體，並將讀取流量路由到主要叢集的複本執行個體。

對於跨區域、唯讀流量，您應該連接到一個次要叢集。建議您以複本集模式使用叢集端點連線至次要叢集。由於所有執行個體都是唯讀複本執行個體，因此您不需要指定讀取偏好設定。若要將延遲降至最低，請選擇您所在地區或離您最近的地區的讀取器端點。

監控 Amazon DocumentDB 全球集群

Amazon DocumentDB (具有 MongoDB 相容性) 與之整合，CloudWatch 讓您可以收集和 분석叢集的操作指標。您可以使用 CloudWatch 主控台、Amazon DocumentDB 主控台、AWS Command Line Interface (AWS CLI) 或 API 監控這些指標 CloudWatch。

若要監視全域叢集，請使用下列 CloudWatch 測量結果。

| 指標 | 描述 |
|--------------------------------|--|
| GlobalClusterReplicatedWriteIO | 從主要叢集磁碟區複製 AWS 區域到叢集磁碟區 (以 5 分鐘為單位) 的平均計費寫入 I/O 作業數 (每 5 分鐘一次 AWS 區域報告)。複製 ReplicatedWriteIOs 到每個次要區域的數目與主要區域 VolumeWriteIOPs 執行的區域內部數目相同。 |
| GlobalClusterDataTransferBytes | 從主叢集傳輸 AWS 區域到次要叢集的資料量 AWS 區域，以位元組為單位。 |
| GlobalClusterReplicationLag | AWS 區域將變更事件從主要叢集複製到次要叢集時的延遲量 (以毫秒為單位) AWS 區域 |

如需如何檢視這些指標的詳細資訊，請參閱 [檢視資 CloudWatch 料](#)。

災難復原和 Amazon DocumentDB 全域叢集

透過使用全域叢集，您可以快速從災難中復原，例如區域故障。從災難復原通常使用 RTO 和 RPO 的值來測量。

- 復原時間點目標 (RTO) — 系統在災難發生後恢復運作狀態所需的時間。換言之，RTO 會測量停機時間。對於全域叢集，RTO 可以按分鐘順序排列。
- 復原點目標 (RPO) — 可能遺失的資料量 (以時間為單位)。對於全域叢集，RPO 通常以秒為單位測量。
- 若要從意外中斷中復原，您可以執行跨區域容錯移轉至全域叢集中的其中一個次要節點。當您的全域叢集具有多個次要區域時，如果主要叢集發 AWS 區域生中斷連線，請確定您卸離所有次要區域。然後，您將其中一個次要區域提升為新的主要區域 AWS 區域。最後，您可以在每個其他次要區域中建立新叢集，並將這些叢集附加到全域叢集。
- 當您將次要叢集提升為主要叢集時，您也需要更新應用程式用來連線到全域叢集的端點。若要從新提升的叢集取得新的寫入器端點，您可以透過移除端點字串中的 `-ro` 來轉換先前的讀取者端點。例如，如果先前的讀取者端點是 `global-16rr-test-cluster-1.cluster-ro-12345678901.us-west-2.docdb.amazonaws.com`，新提升的寫入器端點就會是 `global-16rr-test-cluster-1.cluster-cps2igpwyrrwa.us-west-2.rds.amazonaws.com`。

Amazon DocumentDB 全域叢集的容錯移轉

如果其中一個叢集中的整個叢集 AWS 區域無法使用，您可以升級全域叢集中的另一個叢集以具有讀取/寫入功能。

如果不同 AWS 區域叢集中的叢集比較適合做為主要叢集，您可以手動啟動容錯移轉機制。例如，您可以增加其中一個次要叢集的容量，然後將它提升為主要叢集。或者，AWS 區域可能會改變之間的活動平衡，以便將主要叢集切換到其他叢集 AWS 區域可能會給寫入作業帶來較低的延遲。

下列程序概述如何在 DocumentDB 全域叢集中升級其中一個次要叢集。

若要升級次要叢集：

1. 在中斷的情況下，停止向主要叢集發出 DML 陳述式和其他寫入作業。AWS 區域
2. 從次要叢集識別叢集 AWS 區域以用作新主要叢集。如果您的全域叢集 AWS 區域中有兩個 (或更多) 次要叢集，請選擇延遲時間最短的次要叢集。
3. 從全域叢集中卸離選擇的次要叢集。

從全域叢集移除次要叢集會立即停止從主要叢集到此次要叢集的複寫，並將其升級到具有完整讀取/寫入功能的獨立佈建叢集叢集。與該區域中的主要叢集相關聯的任何其他次要叢集仍然可用，並且可以接受應用程式的呼叫。它們也會取用資源。由於您正在重新建立全域叢集，因此為了避免分割大腦和其他問題，請先移除其他次要叢集，然後按照以下步驟建立新的全域叢集。

如需分離的詳細步驟，請參閱[從 Amazon DocumentDB 全域叢集中移除叢集](#)。

4. 重新設定應用程式，使其使用其新端點，將所有寫入作業傳送至此現在獨立叢集。如果您在建立全域叢集時接受了提供的名稱，則可以從應用程式中的叢集端點字串中移除-ro 來變更端點。

例如，次要叢集的端點my-global.cluster-ro-aaaaabbbbbbb.us-west-1.docdb.amazonaws.com會my-global.cluster-aaaaabbbbbbb.us-west-1.docdb.amazonaws.com在該叢集與全域叢集中分離時變成。

在下一個步驟中，當您開始向其新增區域時，此叢集會成為新全域叢集的主要叢集。

5. 新增AWS 區域至叢集。當您執行這項操作時，從主要到次要的複寫程序即會開始。
6. 視需要新增更多AWS 區域內容，以重新建立支援應用程式所需的拓撲。請確定在進行這些變更之前、期間和之後，將應用程式寫入傳送至正確的叢集，以避免全域叢集中叢集中的資料不一致 (分腦問題)。
7. 當中斷解決後，您準備好再次將原始叢集指派AWS 區域為主要叢集時，請反向執行相同的步驟。
8. 從全域叢集中移除其中一個次要叢集。這將使其能夠為讀/寫流量提供服務。
9. 將所有寫入流量重新導向至原始叢集中的主要叢集AWS 區域。
10. 新增AWS 區域以設定與以前相同AWS 區域的一或多個次要叢集。

Amazon DocumentDB 全域叢集可以使用 AWS SDK 進行管理，讓您建立解決方案以自動化災難復原和業務持續性規劃使用案例的全球叢集容錯移轉程序。我們在 Apache 2.0 授權下為我們的客戶提供了一個這樣的解決方案，並且可以從我們的工具儲存庫在[這裡](#)存取。此解決方案利用 Amazon Route53 進行端點管理，並提供可根據適當事件觸發的 AWS Lambda 函數。

管理 Amazon DocumentDB 集群

若要管理 Amazon 文件資料庫叢集，您必須擁有具有適當 Amazon DocumentDB 控制平面許可的 IAM 政策。這些許可可讓您建立、修改和刪除叢集和執行個體。此 AmazonDocDBFullAccess 策提供管理 Amazon DocumentDB 叢集所需的所有必要許可。

下列主題說明如何在使用 Amazon DocumentDB 叢集時執行各種任務，包括建立、刪除、修改、連線和檢視叢集。

主題

- [了解叢集](#)
- [Amazon DocumentDB 集群設置](#)
- [Amazon DocumentDB 群集存儲配置](#)
- [決定叢集的狀態](#)
- [Amazon DocumentDB 集群生命週](#)
- [擴展 Amazon DocumentDB 集群](#)
- [複製 Amazon DocumentDB 叢集的磁碟區](#)
- [了解 Amazon DocumentDB 叢集容錯](#)

了解叢集

Amazon DocumentDB 可將運算和儲存區隔開來，並將資料複寫和備份卸載到叢集磁碟區。叢集磁碟區提供耐用、可靠且高度可用的儲存層，在三個可用區域以六種方法複寫資料。複本提供了更高的資料可用性與讀取規模調整。每個叢集最多可擴充 15 個複本。

| 名詞 | 描述 | API 操作 (動詞) |
|------|---|---|
| 叢集 | 包含一或多個執行個體，以及一個管理這些執行個體資料的叢集儲存磁碟區。 | create-db-cluster delete-db-cluster describe-db-clusters modify-db-cluster |
| 執行個體 | 從叢集儲存磁碟區讀取資料或將資料寫入其中，都是透過執行個體完成。在指定的叢集，有兩種類型的執行個體：主要執行個體和複寫執行個體。叢集永遠有一個主要執行個體，而且可以有 0-15 個複本。 | create-db-instance delete-db-instance describe-db-instances modify-db-instance describe-orderable-db-instance-options |

| 名詞 | 描述 | API 操作 (動詞) |
|--------|--|--------------------|
| | | reboot-db-instance |
| 叢集磁碟區 | 虛擬資料庫儲存體磁碟區可以橫跨三個可用區域，且每個可用區域都有兩個叢集資料的複本。 | N/A |
| 主要執行個體 | 同時支援讀取和寫入操作，並執行叢集磁碟區所有資料修改。每個叢集具有一個主要執行個體。 | N/A |
| 複本執行個體 | 僅支援讀取操作。除了主執行個體之外，每個 Amazon DocumentDB 叢集最多可以有 15 個複本執行個體。多個複本分配讀取工作負載。透過將複本放置在不同可用區域，您可以提高資料庫可用性。 | N/A |
| 叢集端點 | Amazon DocumentDB 叢集的端點，可連接至叢集的目前主要執行個體。每個 Amazon DocumentDB 叢集都有一個叢集端點和一個主要執行個體。 | N/A |
| 讀取器端點 | Amazon DocumentDB 叢集的端點，可連接至該叢集的其中一個可用複本。每個 Amazon DocumentDB 叢集都有一個讀取器端點。如果有多個複本，讀取器端點會將每個連線請求導向至其中一個 Amazon DocumentDB 複本。 | N/A |

| 名詞 | 描述 | API 操作 (動詞) |
|--------|--|-------------|
| 執行個體端點 | 連線到特定執行個體的 Amazon DocumentDB 叢集中執行個體的端點。叢集中的每個執行個體，不論類型為何，都有自己的唯一執行個體端點。 | N/A |

Amazon DocumentDB 集群設置

當您建立或修改叢集時，請務必了解哪些參數是不可變的以及哪些是在叢集建立後可修改的。下表列出叢集專屬的所有設定或參數。如資料表所指定的，有些可以修改，有些則不可以。

Note

不應將這些設定與 Amazon DocumentDB 叢集參數群組及其參數混淆。如需叢集參數群組的詳細資訊，請參閱 [管理 Amazon DocumentDB 叢集參數群組](#)。

| 參數 | 可修改 | 備註 |
|------------------------------|-----|---|
| DBClusterIdentifier | 是 | 命名限制： <ul style="list-style-type: none"> 長度為 [1—63] 字母、數字或連字號。 第一個字元必須是字母。 不能以連字號結尾或連續包含兩個連字號。 Amazon RDS、Amazon Neptune 和 Amazon DocumentDB 中每 AWS 帳戶個區域的所有叢集必須是唯一的。 |
| Engine | 否 | 必須為 docdb。 |
| BackupRetentionPeriod | 是 | 必須介於 [1-35] 天。 |

| 參數 | 可修改 | 備註 |
|------------------------------------|-----|--|
| DBClusterParameterGroupName | 是 | 命名限制： <ul style="list-style-type: none"> 長度為 [1—255] 個字母數字字元。 第一個字元必須是字母。 不能以連字號結尾或連續包含兩個連字號。 |
| DBSubnetGroupName | 否 | 叢集建立之後，即無法修改叢集的子網路。 |
| EngineVersion | 否 | 值可以是 5.0.0 (預設) 4.0.0、或 3.6.0。 |
| KmsKeyId | 否 | 如果您選擇加密叢集，則無法變更改用來加密叢集的 AWS KMS 金鑰。 |
| MasterUsername | 否 | 叢集建立之後，即無法修改 MasterUsername。 命名限制： <ul style="list-style-type: none"> 長度為 [1—63] 個字母數字字元。 第一個字元必須是字母。 不能是資料庫引擎保留的字。 |
| MasterUserPassword | 是 | 約束： <ul style="list-style-type: none"> 長度為 [8—100] 可列印的 ASCII 字元。 可以使用下列項目以外的任何可列印 ASCII 字元： <ul style="list-style-type: none"> / (正斜線) " (雙引號) @ (@ 符號) |
| Port | 是 | 連接埠號碼適用於在叢集中的所有執行個體。 |
| PreferredBackupWindow | 是 | |

| 參數 | 可修改 | 備註 |
|-----------------------------------|-----|---|
| PreferredMaintenanceWindow | 是 | |
| StorageEncrypted | 否 | 如果您選擇加密叢集，就無法再解密。 |
| StorageType | 是 | <p>資料庫叢集的儲存區類型：標準 (standard) 或 I/O 最佳化 (iopt1)。</p> <p>預設：standard</p> <p>此參數可以使用 <code>CreateDBCluster</code> 和進行配置 <code>ModifyDBCluster</code>。</p> <p>如需詳細資訊，請參閱「Amazon DocumentDB 群集存儲配置」。</p> |
| Tags | 是 | |
| VpcSecurityGroupIds | 否 | 建立叢集之後，您無法修改叢集所在的 VPC。 |

Amazon DocumentDB 群集存儲配置

從 Amazon DocumentDB 5.0 開始，以執行個體為基礎的叢集支援兩種儲存組態類型：

- Amazon DocumentDB 標準儲存：專為具有低至中等 I/O 消耗量的客戶所設計。如果您希望 I/O 成本低於 Amazon DocumentDB 叢集總數的 25%，則此選項可能非常適合您。使用 Amazon DocumentDB 標準儲存組態時，除了執行個體和儲存費用外，還要按 pay-per-request I/O 計費。這表示您的帳單可能會因使用量而有所不同。此組態是針對應用程式不斷變動的 I/O 需求量身打造。
- Amazon DocumentDB I/O 最佳化儲存：專為優先考慮價格可預測性或具有 I/O 密集型應用程式的客戶而設計。I/O 最佳化組態可為需要大量 I/O 工作負載的客戶提供改善的效能、更高的輸送量，並減少延遲。如果您希望 I/O 成本超過 Amazon DocumentDB 叢集總成本的 25%，此選項可提供增強的價格效能。使用 Amazon DocumentDB I/O 最佳化儲存組態，您不需要根據 I/O 操作收費，確保每個帳單週期的可預測成本。此組態可穩定成本，同時提升效能。

您可以每 30 天將現有的資料庫叢集切換一次，切換到 Amazon DocumentDB I/O 最佳化儲存。您可以隨時切換回 Amazon DocumentDB 標準儲存體。您可以使用 AWS CLI 或透過叢集配置頁面 AWS

Management Console 中的 `describe-db-clusters` 指令來追蹤將儲存區組態修改為 I/O 最佳化的下一個日期。

您可以建立包含 Amazon DocumentDB I/O 最佳化組態的新資料庫叢集，也可以在 () 中按幾下 [AWS Management Console](#)、[AWS Command Line Interface \(AWS CLI\)](#) 中的單一參數變更或透過 [SDK](#) 轉換現有的資料庫叢集。[AWS](#) 在修改儲存設定期間或之後，不需要停機或重新啟動執行個體。

| <u>Requirement</u> | <u>Standard</u> | <u>I/O-Optimized</u> | <u>Usage</u> |
|------------------------------|-------------------------------------|-------------------------------------|---|
| Default Storage Type | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| Low to Moderate I/O Workload | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Best if expected I/O charges are less than or equal to 25% |
| Price Predictability | <input type="checkbox"/> | <input checked="" type="checkbox"/> | |
| High I/O Workload | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Best if expected I/O charges are greater than or equal to 25% |
| High Write Throughput | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Average 30%-50% observed improvement |

建立 I/O 最佳化叢集

Using the AWS Management Console

若要建立或修改 I/O 最佳化叢集，請使用：AWS Management Console

- 在 Amazon DocumentDB 管理主控台的叢集下，選擇 [建立] 或選取叢集並選擇 [動作]，然後選擇 [修改]。
- 如果您要建立新叢集，請務必在「叢集類型」段落中選擇「執行處理型叢集」（這是預設選項）。

Cluster type

Instance Based Cluster
Instance based cluster can scale your database to millions of reads per second and up to 64TB of storage capacity. With instance based clusters you can choose your instance type based on your requirements.

Elastic Cluster
Elastic clusters can scale your database to millions of reads and writes per second, with petabytes of storage capacity. Elastic clusters support MongoDB compatible sharding APIs. With Elastic Clusters, you do not need to choose, manage or upgrade instances.

3. 在「組態」區段的「叢集儲存組態」下，選擇「Amazon DocumentDB I/O 最佳化」。

Cluster storage configuration - new [Info](#)

Choose the storage configuration for your Amazon DocumentDB cluster that best fits your application's price predictability and price performance needs.

Storage configuration

Database instance, storage, and I/O charges vary depending on the storage configuration

Amazon DocumentDB Standard

- Pay-per-request I/O charges apply. Instance and storage prices don't include I/O usage.
- Cost-effective pricing for many applications with low to moderate I/O usage.

Amazon DocumentDB I/O-Optimized

- No charges for I/O operations. Instance and storage prices include I/O usage.
- Predictable pricing for all applications. Improved price performance for I/O-intensive applications.

4. 完成叢集建立或修改，然後選擇 [建立叢集] 或 [修改叢集]。

如需完整的「建立叢集」程序，請參閱[使用建立叢集和主要執行個體 AWS Management Console](#)。

如需完整的「修改叢集」程序，請參閱[修改 Amazon DocumentDB 叢集](#)。

Using the AWS CLI

若要使用建立 I/O 最佳化叢集：AWS CLI

在下列範例中，將每個#####取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster \
  --db-cluster-identifier sample-cluster \
  --engine docdb \
  --engine-version 5.0.0 \
  --storage-type iopt1 \
  --deletion-protection \
  --master-username username \
  --master-user-password password
```

針對 Windows：

```
aws docdb create-db-cluster ^
  --db-cluster-identifier sample-cluster ^
  --engine docdb ^
  --engine-version 5.0.0 ^
  --storage-type iopt1 ^
  --deletion-protection ^
```

```
--master-username username ^  
--master-user-password password
```

決定儲存組態的成本分析

使用 Amazon DocumentDB，您可以靈活地為每個擁有的資料庫叢集選擇儲存組態。為了在標準和 I/O 最佳化之間正確配置叢集，您可以逐一追蹤 Amazon DocumentDB 成本。若要這麼做，您可以將標籤新增至現有叢集、在 [AWS Billing and Cost Management 儀表板](#) 中啟用成本配置標記，以及分析中指定叢集的成本 [AWS Cost Explorer Service](#)。有關成本分析的信息，請參閱我們的博客 [使用成本分配標籤](#)。

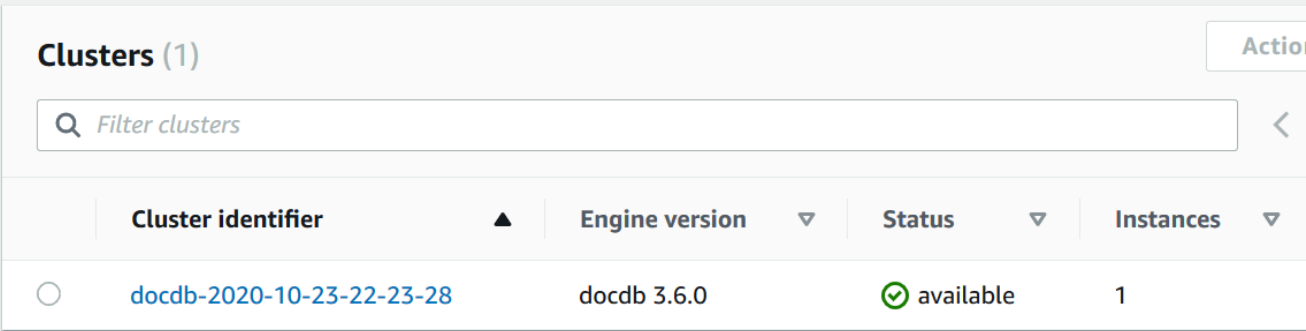
決定叢集的狀態

您可以使用或來判斷叢集的 AWS Management Console 狀態 AWS CLI。

Using the AWS Management Console

請使用下列程序查看 Amazon DocumentDB 叢集的狀態。 AWS Management Console

1. [登入 AWS Management Console](#)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。
3. 在 Cluster identifier (叢集識別碼) 欄中尋找您有興趣之叢集的名稱。然後，閱讀該列上的 Status (狀態) 欄找出叢集的狀態，如下所示。



| Cluster identifier | Engine version | Status | Instances |
|---------------------------|----------------|-----------|-----------|
| docdb-2020-10-23-22-23-28 | docdb 3.6.0 | available | 1 |

Using the AWS CLI

您可以使用此 `describe-db-clusters` 作業查看 Amazon DocumentDB 叢集的狀態，使用 AWS CLI

以下程式碼會尋找叢集 `sample-cluster` 的狀態。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

針對 Windows：

```
aws docdb describe-db-clusters ^  
  --db-cluster-identifier sample-cluster ^  
  --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  [  
    "sample-cluster",  
    "available"  
  ]  
]
```

Amazon DocumentDB 集群生命週

Amazon DocumentDB 叢集的生命週期包括建立、描述、修改和刪除叢集。本節提供如何完成這些程序的相關資訊。

主題

- [創建一個 Amazon DocumentDB 集群](#)
- [描述 Amazon DocumentDB 集群](#)
- [修改 Amazon DocumentDB 叢集](#)
- [判斷擱置的維護](#)
- [對叢集的引擎版本執行修補程式更新](#)
- [停止和啟動 Amazon DocumentDB 集群](#)
- [刪除 Amazon DocumentDB 集群](#)

創建一個 Amazon DocumentDB 集群

Amazon DocumentDB 叢集包含執行個體和代表叢集資料的叢集磁碟區。叢集磁碟區在三個可用區域間以六種方法複寫為單一虛擬磁碟區。叢集包含一個主要執行個體，而且最多可選擇性地包含 15 個複本執行個體。

以下各節說明如何使用 AWS Management Console 或建立 Amazon DocumentDB 叢集。AWS CLI 您可以新增該叢集的其他複本執行個體。當您使用主控台建立 Amazon DocumentDB 叢集時，會同時為您自動建立主要執行個體。如果您使用 AWS CLI 建立 Amazon DocumentDB 叢集，則在叢集狀態可用之後，您必須為該叢集建立主要執行個體。

必要條件

以下是建立 Amazon 文件資料庫叢集的先決條件。

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，[將管理存取權指派給管理使用者](#)，並且僅使用根使用者來執行 [需要根使用者存取權的任務](#)。

VPC 先決條件

您只能在 Amazon Virtual Private Cloud (Amazon VPC) 中創建一個 Amazon DocumentDB 集群。您的 Amazon VPC 必須在至少兩個可用區域中每個子網路都至少有一個子網路，才能將其與 Amazon DocumentDB 叢集搭配使用。透過將叢集執行個體分配到可用區域之間，您可以確保在可用區域發生故障時，叢集中可以使用執行個體。

子網路必備

建立 Amazon DocumentDB 叢集時，您必須在該 VPC 內選擇 VPC 和對應的子網路群組來啟動叢集。子網路決定可用區域，以及在該可用區域內您想要用來啟動執行個體的 IP 範圍。鑒於這項討論的目的，我們將交替使用「子網路」與「可用區域」這兩個名詞。子網路群組是一組具名的子網路 (或「可

用區域」)。子網路群組允許您指定要用於啟動 Amazon DocumentDB 執行個體的可用區域。例如，在具有三個執行個體的叢集中，建議這些執行個體每一個在不同的可用區域佈建高可用性。因此，如果單一可用區域發生故障，它只會影響單一執行個體。

Amazon DocumentDB 執行個體目前最多可在三個可用區域佈建。即使子網路群組有三個以上的子網路，您也只能使用其中三個子網路來建立 Amazon DocumentDB 叢集。因此，建議您在建立子網路群組時，只選擇您要部署執行個體的三個子網路。在美國東部 (維吉尼亞北部)，您的子網路群組可以有六個子網路 (或可用區域)。不過，當佈建 Amazon DocumentDB 叢集時，Amazon DocumentDB 會選擇其中三個用於佈建執行個體的可用區域。

例如，假設您在建立叢集時，Amazon DocumentDB 會選擇可用區域 {1A、1B 和 1C}。如果您嘗試在可用區域 {1D} 中建立執行個體，API 呼叫會失敗。但是，如果您選擇在不指定特定可用區域的情況下建立執行個體，Amazon DocumentDB 會代表您選擇一個可用區域。Amazon DocumentDB 使用演算法在可用區域之間對執行個體進行負載平衡，以協助您達到高可用性。例如，若佈建三個執行個體，預設會跨三個可用區域佈建執行個體，而不會全佈建在單一可用區域。

建議：

- 除非有特定的原因，否則一律使用三個子網路建立子網路群組。這樣做可協助確保具有三個或更多個執行個體的叢集能夠獲得更高的可用性，因為可以跨三個可用區域佈建執行個體。
- 一律將執行個體分散在多個可用區域以達到高可用性。請勿將叢集的所有執行個體放置在單一可用區域。
- 由於隨時都會發生容錯移轉事件，您不應假設主要執行個體或複本執行個體一律位在特定的可用區域。

其他先決條件

以下是建立 Amazon DocumentDB 叢集的一些其他先決條件：

- 如果您要連線到 AWS 使用 AWS Identity and Access Management (IAM) 登入資料，您的 IAM 帳戶必須具有 IAM 政策，以授與執行 Amazon DocumentDB 操作所需的許可。

如果您使用 IAM 帳戶存取 Amazon DocumentDB 主控台，則必須先使用 IAM 帳戶登入。AWS Management Console [然後轉到 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>](https://console.aws.amazon.com/docdb)。

- 如果您想要為叢集量身打造組態參數，則必須為叢集參數群組及參數群組指定必要的參數設定。如需建立或修改叢集參數群組或參數群組的相關資訊，請參閱[管理 Amazon DocumentDB 叢集參數群組](#)。

- 您必須決定要為叢集指定的 TCP/IP 連接埠號碼。某些公司的防火牆會封鎖與 Amazon DocumentDB 預設連接埠的連線。如果您公司的防火牆會封鎖預設連接埠，請為您的叢集選擇另一個連接埠。叢集中的所有執行個體都使用相同的連接埠。

使用建立叢集和主要執行個體 AWS Management Console

下列程序說明如何使用主控台啟動具有一或多個執行個體的 Amazon DocumentDB 叢集。

建立叢集：使用預設設定

使用預設設定建立具有執行處理的叢集 AWS Management Console

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 如果您要在美國東部 (維吉尼亞北部) 區域以外的 AWS 區域 其他區域建立叢集，請從主控台右上角區段的清單中選擇「區域」。
3. 在導覽窗格中，選擇 Clusters (叢集)，然後選擇 Create (建立)。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示

(☰)

)。

4. 在 [建立 Amazon DocumentDB 叢集] 頁面上，完成設定窗格。
 - a. 叢集識別碼 — 接受 Amazon DocumentDB 提供的名稱，或輸入叢集的名稱；例如，`sample-cluster`

叢集命名限制條件：

- 長度為 [1—63] 字母、數字或連字號。
 - 第一個字元必須是字母。
 - 不能以連字號結尾或連續包含兩個連字號。
 - Amazon RDS、Neptune 和亞馬 Amazon DocumentDB 中每 AWS 帳戶個區域的所有叢集必須是唯一的。
- b. 引擎版本 — 接受預設引擎版本 4.0.0，或選擇性地選擇 3.6.0。
 - c. 實體類別 — 接受預設值 `db.r5.large`，或從清單中選擇您要的實體類別。

- d. 執行個體數目 — 在清單中，選擇要使用此叢集建立的執行個體數目。第一個執行個體將是主要執行個體，而其他所有執行個體將是唯讀複本執行個體。如有必要，您可以稍後新增和刪除執行個體。根據預設，Amazon DocumentDB 叢集會以三個執行個體 (一個主要複本和兩個複本) 啟動。
5. 完成 [叢集儲存配置] 區段。

選擇 Amazon DocumentDB 標準 (默認) 或 Amazon DocumentDB I/O 優化。如需詳細資訊，請參閱 [Amazon DocumentDB 群集存儲配置](#)。

6. 完成 Authentication (認證) 窗格。
 - a. 使用者名稱 — 輸入主要使用者的名稱。若要登入叢集，您必須使用主要使用者名稱。

主要使用者命名限制：
 - 長度為 [1—63] 個字母數字字元。
 - 第一個字元必須是字母。
 - 不能是資料庫引擎保留的字。
 - b. 密碼 — 輸入主要使用者的密碼，然後加以確認。若要登入叢集，您必須使用主要使用者的密碼。

密碼限制條件：

- 長度為 [8-100] 個可列印的 ASCII 字元。
- 可以使用下列項目以外的任何可列印 ASCII 字元：
 - / (正斜線)
 - " (雙引號)
 - @ (@ 符號)

7. 在螢幕下方，選擇以下其中一項：
 - 若要立即建立叢集，請選擇 Create cluster (建立叢集)。
 - 若不建立叢集，請選擇 Cancel (取消)。
 - 若要在建立之前進一步設定叢集，請選擇 Show additional configurations (顯示其他組態)，然後繼續進行[建立叢集：其他組態](#)。

Additional Configurations (其他組態) 部分涵蓋的組態如下所示：

- 網路設定 — 預設值為使用 default VPC 安全性群組。

- 叢集選項 — 預設為使用連接埠為 27017 且預設參數群組。
- 加密-預設值是啟用使用(default) aws/rds金鑰加密。

⚠ Important

叢集加密後，就無法解密。

- Backup — 預設值是保留備份 1 天，並讓 Amazon DocumentDB 選擇備份時段。
- 防護記錄匯出 — 預設值是不將稽核記錄匯出至防 CloudWatch 護記錄。
- 維護 — 預設是讓 Amazon DocumentDB 選擇維護時段。
- 刪除保護：保護叢集免於意外刪除。使用主控台建立的叢集預設為已啟用。

如果現在接受預設設定，您可以在稍後藉由修改叢集來將其大多數做變更。

8. 為叢集的安全群組啟用入站連線。

如果您並未變更叢集的預設值，表示您使用預設安全群組為指定區域中的預設 VPC 建立叢集。若要連線到 Amazon DocumentDB，您必須為叢集的安全群組啟用連接埠 27017 (或您選擇的連接埠) 上的輸入連線。

新增入站連線至叢集安全群組

- 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
- 在主視窗的 Resources (資源) 區段中，選擇 Security groups (安全群組)。

Resources

You are using the following Amazon EC2 resources in the EU West (Ireland) region:

| | |
|---------------------|-------------------|
| 0 Running Instances | 0 Elastic IPs |
| 0 Dedicated Hosts | 0 Snapshots |
| 0 Volumes | 0 Load Balancers |
| 0 Key Pairs | 1 Security Groups |
| 0 Placement Groups | |

- 從安全群組清單中，找出您建立叢集時使用的安全群組 (最可能是預設安全群組)，並選擇安全群組名稱左側的方塊。

| <input type="checkbox"/> | Name | Group ID | Group Name | VPC ID |
|-------------------------------------|------|----------------------|------------|--------------|
| <input checked="" type="checkbox"/> | | sg-06b2ad61 | default | vpc-d833a4bc |
| <input type="checkbox"/> | | sg-07443a112c70a5282 | test-sg | vpc-d833a4bc |

- d. 在 Actions (動作) 功能表中，選擇 Edit inbound rules (編輯入站規則)，然後輸入規則限制。
 - i. 類型 — 從清單中，選擇要開啟給網路流量的通訊協定。
 - ii. 通訊協定 — 從清單中選擇通訊協定的類型。
 - iii. 連接埠範圍 — 對於自訂規則，輸入連接埠號碼或連接埠範圍。請確定連接埠號碼或範圍包括您建立叢集時指定的連接埠 (預設：27107)。
 - iv. 來源：指定可以連線到您執行個體的流量。從清單中選擇流量來源。如果您選擇 Custom (自訂)，指定單一 IP 地址或是以 CIDR 表示法表示的 IP 地址範圍 (例如 203.0.113.5/32)。
 - v. 說明 — 輸入此規則的說明。
 - vi. 完成建立規則後，選擇 Save (儲存)。

建立叢集：其他組態

如果您想要接受叢集的預設設定，您可以略過以下步驟，然後選擇 Create cluster (建立叢集)。

1. 完成 Network settings (網路設定) 窗格。

Network settings

a Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this cluster.

vpc-91280df6

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

b Subnet group [Info](#)
A subnet group is a collection of subnets that are within a VPC.

default

c VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups

default (VPC) X

- a. Virtual Private Cloud (VPC) (VPC) — 在清單中，選擇要在其中啟動此叢集的 Amazon VPC。
 - b. 子網路群組 — 在清單中，選擇要用於此叢集的子網路群組。
 - c. VPC 安全群組 — 在清單中，選擇此叢集的 VPC 安全性群組。
2. 完成 Cluster options (叢集選項) 窗格。

Cluster options

Port
TCP/IP port that is used to connect to the cluster.

Cluster parameter group [Info](#)

- a. 資料庫連接埠 — 使用向上和向下箭號來設定應用程式用來連線至執行個體的 TCP/IP 連接埠。
 - b. 叢集參數群組 — 在參數群組清單中，選擇此叢集的叢集參數群組。
3. 完成 Encryption (加密) 窗格。

Encryption-at-rest

Encryption-at-rest [Info](#)

Enable encryption
 Disable encryption

AWS KMS Key

Account

713738290397

KMS key ID

32d28de3-8254-4597-a3da-571ddc95b76f

- a. Encryption-at-rest-選擇下列其中一項：
 - 啟用加密 — 預設。所有靜態資料已加密。如果您選擇加密資料，您無法復原此動作。
 - 停用加密 — 您的資料未加密。
- b. AWS KMS 金鑰 — 只有在您正在加密資料時才能使用此功能。在清單中，選擇要用於在此叢集中加密的金鑰。預設值為 (default) aws/rds。

如果您選擇 Enter a key ARN (輸入金鑰 ARN)，您必須需要輸入金鑰的 Amazon Resource Name (ARN)。

4. 完成 Backup (加密) 窗格。

- a. Backup 保留期間：在清單中，選擇在刪除叢集之前保留自動備份的天數。
- b. Backup 時段 — 設定 Amazon DocumentDB 要備份此叢集的每日時間和持續時間。
 - i. 開始時間 — 在第一個清單中，選擇開始時間小時 (UTC) 以開始自動備份。在第二個清單中，選擇您要自動備份開始進行的時間 (分)。
 - ii. 持續時間 — 在清單中，選擇要分配給建立自動備份的小時數。

5. 選取要匯出至記錄檔的記錄類型，以完成 [記錄 CloudWatch 檔匯出] 窗格。

- 稽核日誌 — 選取此選項以允許將稽核日誌匯出到 Amazon CloudWatch 日誌。如果您選取 Audit logs (稽核日誌)，則必須在叢集的自訂參數群組中啟用 audit_logs。如需詳細資訊，請參閱 [稽核 Amazon DocumentDB 事件](#)。
- 效能分析工具記錄 — 選取此選項可讓您將作業效能分析工具日誌匯出至 Amazon 日誌。CloudWatch 如果您選取 Profiler logs (Profiler 日誌)，則還必須在叢集的自訂參數群組中修改下列參數：
 - profiler 設定為 enabled
 - profiler_threshold_ms 設定為值 [0-INT_MAX] 以設定效能分析作業的臨界值。
 - profiler_sampling_rate 設定為值以設 [0.0-1.0] 定輪廓慢速操作的百分比。

如需詳細資訊，請參閱 [分析亞馬遜文檔數據庫操作](#)。

6. 完成 Maintenance (維護) 窗格。

The screenshot shows the 'Maintenance' window. At the top left, there is a red circle with the letter 'a' next to the 'Maintenance window' header. Below the header, there is a sub-header 'Maintenance window Info' and a description: 'The period in which pending modifications or patches are applied to Instances in the cluster.' There are two radio buttons: 'Select window' (which is selected) and 'No preference'. Below these are three dropdown menus: 'Start day' (set to 'Monday'), 'Start time' (set to '00 : 00 UTC'), and 'Duration' (set to '0.5 hours').

- 選擇下列其中一項
 - 選取時段 — 您可以指定 Amazon DocumentDB 在叢集上執行維護的星期幾、UTC 開始時間和持續時間。
 - a. 開始日期 — 在清單中，選擇要啟動叢集維護的星期幾。
 - b. 開始時間 — 在清單中，選擇要開始維護的小時和分鐘 (UTC)。
 - c. 持續時間 — 在清單中，選擇為叢集維護配置多少時間。如果無法在指定時間內完成維護作業，維護程序會持續超過指定的時間，直到完成為止。
 - 無偏好設定 — Amazon DocumentDB 會選擇星期幾、開始時間和執行維護的持續時間。

7. 如果您希望將一或多個標籤新增到此叢集，請完成 Tags (標籤) 窗格。

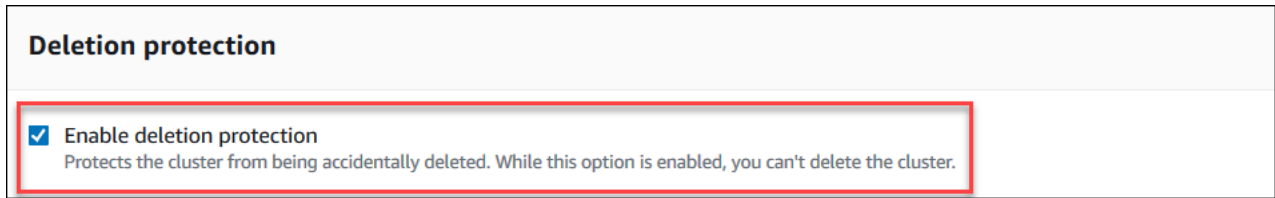
The screenshot shows the 'Tags' window. At the top left, there is a red circle with the letter 'a' next to the 'Add tag' button. Above the input fields, there are two red circles with letters 'b' and 'c'. 'b' is next to the 'Key' label, and 'c' is next to the 'Value - optional' label. Below these are two input fields: 'Enter key' and 'Enter value', followed by a 'Remove tag' button.

對於您要新增到叢集的每個標籤，請重複下列步驟。一個叢集上最多可有 10 個。

- a. 選擇 Add tags (新增標籤)。
- b. 輸入標籤的 Key (索引鍵)。
- c. 選擇性輸入標籤的 Value (值)。

若要移除標籤，請選擇 Remove tag (移除標籤)。

- 當您使用主控台建立叢集時，預設會啟用 Deletion Protection (刪除保護)。若要停用刪除保護，請清除 Enable deletion protection (啟用刪除保護)。若已啟用，刪除保護就會避免叢集遭到刪除。若要刪除已啟用刪除保護的叢集，您必須先修改叢集以停用刪除保護。



如需刪除保護的詳細資訊，請參閱 [刪除 Amazon DocumentDB 集群](#)。

- 若要建立叢集，請選擇 Create cluster (建立叢集)。否則，請選擇 Cancel (取消)。

使用建立叢集 AWS CLI

下列程序說明如何使用啟動 Amazon DocumentDB 叢集和建立 Amazon DocumentDB 複本。AWS CLI

參數

- **--db-cluster-identifier** - 必要項目。識別此叢集的小寫字串。

叢集命名限制條件：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- 每個 AWS 帳戶、每個區域的所有叢集 (跨 Amazon RDS、亞馬遜海王星和 Amazon DocumentDB) 必須是唯一的。
- **--engine** - 必要項目。必須為 **docdb**。
- **--deletion-protection** | **--no-deletion-protection** 選擇性。若已啟用刪除保護，它就會避免叢集遭到刪除。使用時 AWS CLI，預設設定為停用刪除保護。

如需刪除保護的詳細資訊，請參閱 [刪除 Amazon DocumentDB 集群](#)。

- **--storage-type standard | iopt1** 選擇性。預設：**standard**。叢集的儲存配置。有效值為 **standard** (標準) 或 **iopt1** (I/O 最佳化)。
- **--master-username** - 必要項目。用於驗證使用者的使用者名稱。

主要使用者命名限制條件：

- 長度為 [1-63] 英數字元。
 - 第一個字元必須是字母。
 - 不能是資料庫引擎保留的字。
- **--master-user-password** - 必要項目。用於驗證使用者的使用者密碼。

主要密碼限制條件：

- 長度為 [8-100] 個可列印的 ASCII 字元。
- 可以使用下列項目以外的任何可列印 ASCII 字元：
 - / (正斜線)
 - " (雙引號)
 - @ (@ 符號)

如需其他參數的詳細資訊，請參閱[CreateDBCluster](#)。

若要使用啟動 Amazon DocumentDB 叢集 AWS CLI

若要建立 Amazon DocumentDB 叢集，請呼叫 `create-db-cluster` AWS CLI 下列 AWS CLI 命令會建立一個名為 `sample-cluster` 用刪除保護的 Amazon DocumentDB 叢集。如需刪除保護的詳細資訊，請參閱[刪除 Amazon DocumentDB 集群](#)。

此外，這 `--engine-version` 是一個可選參數，默認為最新的主要引擎版本。目前的主要引擎版本為 4.0.0。發行新的主要引擎版本時，的預設引擎版本 `--engine-version` 將會更新，以反映持續的主要引擎版本。因此，對於生產工作負載，尤其是依賴指令碼、自動化或 AWS CloudFormation 範本的工作負載，我們建議您明確指定 `--engine-version` 預期的主要版本。

Note

如果未指定 `db-subnet-group-name` 或 `vpc-security-group-id`，Amazon DocumentDB 將使用指定區域的預設子網路群組和 Amazon VPC 安全群組。

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --engine docdb \  
  --engine-version 4.0.0 \  
  --deletion-protection \  
  --master-username masteruser \  
  --master-user-password password
```

針對 Windows：

```
aws docdb create-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --engine docdb ^  
  --engine-version 4.0.0 ^  
  --deletion-protection ^  
  --master-username masteruser ^  
  --master-user-password password
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBCluster": {  
    "StorageEncrypted": false,  
    "DBClusterMembers": [],  
    "Engine": "docdb",  
    "DeletionProtection" : "enabled",  
    "ClusterCreateTime": "2018-11-26T17:15:19.885Z",  
    "DBSubnetGroup": "default",  
    "EngineVersion": "4.0.0",  
    "MasterUsername": "masteruser",  
    "BackupRetentionPeriod": 1,  
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",  
    "DBClusterIdentifier": "sample-cluster",  
    "MultiAZ": false,
```

```
"DBClusterParameterGroup": "default.docdb4.0",
"PreferredBackupWindow": "09:12-09:42",
"DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVLNTU7XVAY",
"PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
"Port": 27017,
"Status": "creating",
"ReaderEndpoint": "sample-cluster.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
"AssociatedRoles": [],
"HostedZoneId": "ZNKXTT8WH85VW",
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-77186e0d",
    "Status": "active"
  }
],
"AvailabilityZones": [
  "us-east-1a",
  "us-east-1c",
  "us-east-1e"
],
"Endpoint": "sample-cluster.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
}
```

建立叢集需要幾分鐘才能完成。您可以使用 AWS Management Console 或 AWS CLI 來監視叢集的狀態。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 集群的狀態](#)。

Important

當您使用建立 Amazon DocumentDB 叢集時，不會建立任何執行個體。AWS CLI 因此，您必須明確建立主要執行個體，以及您需要的任何複本執行個體。您可以使用主控台或 AWS CLI 建立執行個體。如需詳細資訊，請參閱 [將 Amazon DocumentDB 執行個體新增至叢集](#)。

如需詳細資訊，請參閱 [CreateDBCluster](#) 亞 Amazon DocumentDB API 參考中的。

描述 Amazon DocumentDB 集群

您可以使用 Amazon DocumentDB 管理主控台或查看與 Amazon DocumentDB 叢集相關的詳細資訊，例如連線端點、安全群組、VPC 和參數群組。AWS CLI

如需詳細資訊，請參閱下列內容：

- [監控 Amazon DocumentDB 集群的狀態](#)
- [尋找叢集端點](#)

Using the AWS Management Console

請使用下列程序，使用主控台檢視指定 Amazon DocumentDB 叢集的詳細資料。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示

(☰)

)。

3. 在叢集清單中，選擇想要檢視詳細資訊的叢集名稱。叢集的相關資訊分為下列群組：
 - 摘要 — 叢集的一般資訊，包括引擎版本、叢集狀態、擱置維護，以及其參數群組的狀態。
 - 連線與安全性 — 「Connect 線」區段會列出使用 mongo Shell 或應用程式連線至此叢集的連線端點。Security Groups (安全群組) 區段會列出與此叢集相關聯的安全群組，以及其 VPC ID 和描述。
 - 組態 — 叢集詳細資料區段列出有關叢集的詳細資訊，包括叢集的 Amazon 資源名稱 (ARN)、端點和參數群組。它也會列出叢集的備份資訊、維護詳細資訊，以及安全性和網路設定。Cluster instances (叢集執行個體) 區段會列出屬於您此叢集的執行個體，當中每個執行個體的角色和叢集參數群組狀態。
 - 監控 — 此叢集的 Amazon CloudWatch 日誌指標。如需詳細資訊，請參閱 [使用監控亞馬遜文檔數據庫 CloudWatch](#)。
 - 事件與標籤 — [最近的事件] 區段會列出此叢集的最近事件。Amazon DocumentDB 會保留與叢集、執行個體、快照、安全群組和叢集參數群組相關的事件記錄。此資訊包括與每個事件相關聯的日期、時間和訊息。Tags (標籤) 區段會列出連接至此叢集的標籤。

Using the AWS CLI

若要使用檢視 Amazon DocumentDB 叢集的詳細資料 AWS CLI，請使用下 `describe-db-clusters` 列範例所示的命令。如需詳細資訊，請參閱 Amazon DocumentDB 源管理 API 參考 [DescribeDBClusters](#) 中的。

Note

對於某些管理功能，例如叢集和執行個體生命週期管理，Amazon DocumentDB 會運用與 Amazon RDS 共用的操作技術。 `filterName=engine,Values=docdb` 篩選器參數只會傳回 Amazon DocumentDB 叢集。

Example

範例 1：列出所有 Amazon DocumentDB 叢集

下列 AWS CLI 程式碼列出區域中所有 Amazon DocumentDB 叢集的詳細資料。

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

此操作的輸出將會如下所示。

```
{
  "DBClusters": [
    {
      "AvailabilityZones": [
        "us-east-1c",
        "us-east-1b",
        "us-east-1a"
      ],
      "BackupRetentionPeriod": 1,
      "DBClusterIdentifier": "sample-cluster-1",
      "DBClusterParameterGroup": "sample-parameter-group",
      "DBSubnetGroup": "default",
      "Status": "available",
      ...
    },
    {
      "AvailabilityZones": [
        "us-east-1c",
        "us-east-1b",
```

```

        "us-east-1a"
    ],
    "BackupRetentionPeriod": 1,
    "DBClusterIdentifier": "sample-cluster-2",
    "DBClusterParameterGroup": "sample-parameter-group",
    "DBSubnetGroup": "default",
    "Status": "available",
    ...
},
{
    "AvailabilityZones": [
        "us-east-1c",
        "us-east-1b",
        "us-east-1a"
    ],
    "BackupRetentionPeriod": 1,
    "DBClusterIdentifier": "sample-cluster-3",
    "DBClusterParameterGroup": "sample-parameter-group",
    "DBSubnetGroup": "default",
    "Status": "available",
    ...
}
]
}

```

Example

範例 2：列出指定的 Amazon 文件資料庫叢集的所有詳細資料

下列 AWS CLI 程式碼會列出叢集的詳細資訊 sample-cluster。

若為 Linux、macOS 或 Unix：

```

aws docdb describe-db-clusters \
  --filter Name=engine,Values=docdb \
  --db-cluster-identifier sample-cluster

```

針對 Windows：

```

aws docdb describe-db-clusters ^
  --filter Name=engine,Values=docdb ^
  --db-cluster-identifier sample-cluster

```

此操作的輸出將會如下所示。

```
{
  "DBClusters": [
    {
      "AllocatedStorage": 1,
      "AvailabilityZones": [
        "us-east-1c",
        "us-east-1a",
        "us-east-1d"
      ],
      "BackupRetentionPeriod": 2,
      "DBClusterIdentifier": "sample-cluster",
      "DBClusterParameterGroup": "sample-parameter-group",
      "DBSubnetGroup": "default",
      "Status": "available",
      "EarliestRestorableTime": "2023-11-07T22:34:08.148000+00:00",
      "Endpoint": "sample-cluster.node.us-east-1.amazon.com",
      "ReaderEndpoint": "sample-cluster.node.us-east-1.amazon.com",
      "MultiAZ": false,
      "Engine": "docdb",
      "EngineVersion": "5.0.0",
      "LatestRestorableTime": "2023-11-10T07:21:16.772000+00:00",
      "Port": 27017,
      "MasterUsername": "chimeraAdmin",
      "PreferredBackupWindow": "22:22-22:52",
      "PreferredMaintenanceWindow": "sun:03:01-sun:03:31",
      "ReadReplicaIdentifiers": [],
      "DBClusterMembers": [
        {
          "DBInstanceIdentifier": "sample-instance-1",
          "IsClusterWriter": true,
          "DBClusterParameterGroupStatus": "in-sync",
          "PromotionTier": 1
        },
        {
          "DBInstanceIdentifier": "sample-instance-2",
          "IsClusterWriter": true,
          "DBClusterParameterGroupStatus": "in-sync",
          "PromotionTier": 1
        }
      ],
      "VpcSecurityGroups": [
```

```

        {
            "VpcSecurityGroupId": "sg-9084c2ec",
            "Status": "active"
        }
    ],
    "HostedZoneId": "Z06853723JYKYBXTJ49RB",
    "StorageEncrypted": false,
    "DbClusterResourceId": "cluster-T4LGLANHVAPGQYYULWUDKLVQL4",
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "ClusterCreateTime": "2023-11-06T18:05:41.568000+00:00",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "CopyTagsToSnapshot": false,
    "CrossAccountClone": false,
    "DomainMemberships": [],
    "TagList": [],
    "StorageType": "iopt1",
    "AutoMinorVersionUpgrade": false,
    "NetworkType": "IPV4",
    "IOOptimizedNextAllowedModificationTime":
"2023-12-07T18:05:41.580000+00:00"
    }
]
}

```

Example

範例 3：列出 Amazon 文件資料庫叢集的特定詳細資料

若要使用列出叢集詳細資訊的子集 AWS CLI，請新增一個來指 `--query` 定要列出 `describe-db-clusters` 作業的叢集成員。 `--db-cluster-identifier` 參數是您想要顯示詳細資訊之特定叢集的識別碼。如需有關查詢的詳細資訊，請參閱 [《AWS Command Line Interface 使用指南》中的如何使用 `--query` 選項篩選輸出](#)。

下列範例列出 Amazon DocumentDB 叢集中的執行個體。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \
```



```
--filter Name=engine,Values=docdb \  
--db-cluster-identifier sample-cluster \  
--query 'DBClusters[*].[DBClusterMembers]'
```

針對 Windows：

```
aws docdb describe-db-clusters ^  
--filter Name=engine,Values=docdb ^  
--db-cluster-identifier sample-cluster ^  
--query 'DBClusters[*].[DBClusterMembers]'
```

此操作的輸出將會如下所示。

```
[  
  [  
    [  
      {  
        "DBInstanceIdentifier": "sample-instance-1",  
        "IsClusterWriter": true,  
        "DBClusterParameterGroupStatus": "in-sync",  
        "PromotionTier": 1  
      },  
      {  
        "DBInstanceIdentifier": "sample-instance-2",  
        "IsClusterWriter": false,  
        "DBClusterParameterGroupStatus": "in-sync",  
        "PromotionTier": 1  
      }  
    ]  
  ]  
]
```

修改 Amazon DocumentDB 叢集


若要修改叢集，叢集必須處於可用狀態。您無法修改已停止的叢集。如果叢集已停止，請先啟動叢集，等待叢集變成可用，然後進行所需的修改。如需詳細資訊，請參閱 [停止和啟動 Amazon DocumentDB 集群](#)。

Using the AWS Management Console

使用下列程序來使用主控台修改特定的 Amazon DocumentDB 叢集。

若要修改亞 Amazon DocumentDB 叢集

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

 Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 選擇叢集名稱左側的按鈕，以指定您要修改的叢集。
4. 選擇 Actions (動作)，然後選擇 Modify (修改)。
5. 在 Modify Cluster: <cluster-name> (修改叢集: <叢集-名稱>) 窗格中，進行您要的變更。您可以在以下領域進行變更：
 - 叢集規格 — 叢集的名稱、安全群組和密碼。
 - 叢集儲存配置 — 叢集的資料儲存模式。在「標準」和「I/O 最佳化」組態之間選擇。
 - 叢集選項 — 叢集的連接埠和參數群組。
 - Backup — 叢集的備份保留期和備份時段。
 - 記錄匯出 — 啟用或停用匯出稽核或效能分析工具記錄。
 - 維護 — 設定叢集的維護時段。
 - 刪除保護 — 啟用或停用叢集上的刪除保護。預設會啟用刪除保護。
6. 完成後，請選擇 Continue (繼續) 以檢視變更的摘要。
7. 如果您滿意變更，可以選擇 Modify cluster (修改叢集) 來修改叢集。或者，您也可以選擇 Back (返回) 或 Cancel (取消)，分別編輯或取消變更。

套用您的變更需要幾分鐘的時間才會完成。您只能使用狀態為可用的叢集。您可以使用主控台或 AWS CLI 監控叢集的狀態。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 集群的狀態](#)。

Using the AWS CLI

使用 AWS CLI 透過 `modify-db-cluster` 操作來修改指定的叢集。如需詳細資訊，請參閱 [ModifyDBCluster](#) 亞 Amazon DocumentDB API 參考中的。

參數

- **--db-cluster-identifier** - 必要項目。您要修改之 Amazon DocumentDB 叢集的識別碼。
- **--backup-retention-period** 選擇性。自動備份保留的天數。有效值為 1—35。
- **--storage-type** 選擇性。叢集的儲存配置。有效值為 `standard` (標準) 或 `iopt1` (I/O 最佳化)。
- **--db-cluster-parameter-group-name** 選擇性。要用於叢集的叢集參數群組名稱。
- **--master-user-password** 選擇性。主要資料庫使用者的新密碼。

密碼限制條件：

- 長度為 [8—100] 可列印的 ASCII 字元。
- 可以使用下列項目以外的任何可列印 ASCII 字元：
 - / (正斜線)
 - " (雙引號)
 - @ (@ 符號)
- **--new-db-cluster-identifier** 選擇性。重新命名叢集時，叢集的新叢集識別碼。此值會以小寫字母字串的形式儲存。

命名限制：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- 對於跨 Amazon RDS、亞馬遜 Amazon Neptune 和 Amazon DocumentDB 的所有叢集，每 AWS 帳戶個區域都必須是唯一的。
- **--preferred-backup-window** 選擇性。每日時間範圍，自動備份會在此期間建立，以國際標準時間 (UTC) 表示。
 - 格式：`hh24:mm-hh24:mm`
- **--preferred-maintenance-window** 選擇性。系統維護可能發生的每週時間範圍，以 UTC 表示。

- 格式：ddd:hh24:mm-ddd:hh24:mm
- 有效天：Sun、Mon、Tue、Wed、Thu、Fri 和 Sat。
- **--deletion-protection**或 **--no-deletion-protection**可選。是否應在此叢集上啟用刪除保護。在叢集修改成停用刪除保護前，刪除保護可避免意外刪除叢集。如需詳細資訊，請參閱 [刪除 Amazon DocumentDB 集群](#)。
- **--apply-immediately**或 **--no-apply-immediately** — 用**--apply-immediately**於立即進行變更。使用 **--no-apply-immediately** 可在叢集下一次維護時段進行變更。

Example

以下程式碼會變更 `sample-cluster` 叢集的備份保留期。

若為 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --apply-immediately \  
  --backup-retention-period 7
```

針對 Windows：

```
aws docdb modify-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --apply-immediately ^  
  --backup-retention-period 7
```

此操作的輸出將會如下所示。

```
{  
  "DBCluster": {  
    "BackupRetentionPeriod": 7,  
    "DbClusterResourceId": "cluster-VDP53QEWST7YHM36TTX0PJT5YE",  
    "Status": "available",  
    "DBClusterMembers": [  
      {  
        "PromotionTier": 1,  
        "DBClusterParameterGroupStatus": "in-sync",  
        "DBInstanceIdentifier": "sample-cluster-instance",  
        "IsClusterWriter": true
```

```

    }
  ],
  "ReadReplicaIdentifiers": [],
  "AvailabilityZones": [
    "us-east-1b",
    "us-east-1c",
    "us-east-1a"
  ],
  "ReaderEndpoint": "sample-cluster.cluster-ro-ctevjxdlur57.us-
east-1.rds.amazonaws.com",
  "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
  "PreferredMaintenanceWindow": "sat:09:51-sat:10:21",
  "EarliestRestorableTime": "2018-06-17T00:06:19.374Z",
  "StorageEncrypted": false,
  "MultiAZ": false,
  "AssociatedRoles": [],
  "MasterUsername": "<your-master-user-name>",
  "DBClusterIdentifier": "sample-cluster",
  "VpcSecurityGroups": [
    {
      "Status": "active",
      "VpcSecurityGroupId": "sg-77186e0d"
    }
  ],
  "HostedZoneId": "Z2SUY0A1719RZT",
  "LatestRestorableTime": "2018-06-18T21:17:05.737Z",
  "AllocatedStorage": 1,
  "Port": 27017,
  "Engine": "docdb",
  "DBClusterParameterGroup": "default.docdb3.4",
  "Endpoint": "sample-cluster.cluster-ctevjxdlur57.us-
east-1.rds.amazonaws.com",
  "DBSubnetGroup": "default",
  "PreferredBackupWindow": "00:00-00:30",
  "EngineVersion": "3.4",
  "ClusterCreateTime": "2018-06-06T19:25:47.991Z",
  "IAMDatabaseAuthenticationEnabled": false
}
}

```

套用您的變更需要幾分鐘的時間才會完成。您只能使用狀態為可用的叢集。您可以使用主控台或 AWS CLI 監控叢集的狀態。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 集群的狀態](#)。

判斷擱置的維護

您可以判斷是否有擱置中的叢集維護，以判斷是否擁有最新的 Amazon DocumentDB 引擎版本。

Using the AWS Management Console

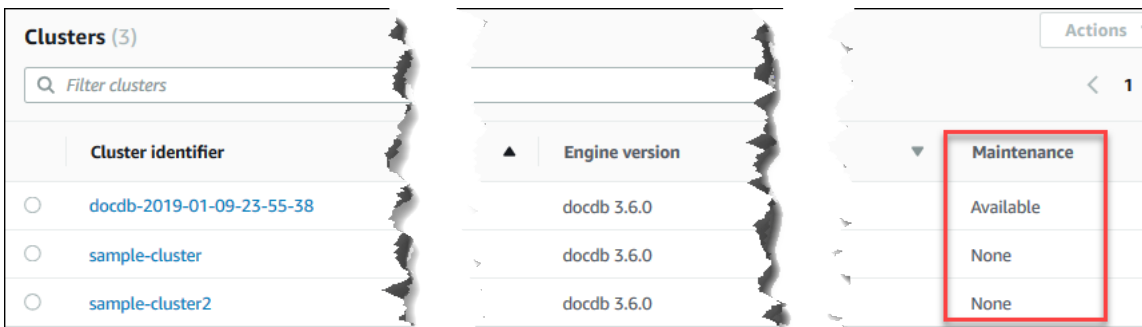
您可以使用 AWS Management Console 來判斷叢集是否有擱置中的維護。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 找出 Maintenance (維護) 欄，判斷叢集是否有待定的維護。



None (無) 表示叢集執行的是最新的引擎版本。Available (可用) 表示叢集有待定的維護，這可能表示引擎需要升級。

4. 如果您的叢集有待定的維護，請繼續[對叢集的引擎版本執行修補程式更新](#)中的步驟。

Using the AWS CLI

您可以使用具有下列參數的 describe-pending-maintenance-actions 作業 AWS CLI 來判斷叢集是否具有最新的引擎版本。

參數

- **--resource-identifier** 選擇性。資源的 ARN (叢集)。如果省略這個參數，則會列出所有叢集的待定維護動作。

- **--region**選擇性。您希望執行此操作的 AWS 區域，例如 us-east-1。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb describe-pending-maintenance-actions \  
  --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster \  
  --region us-east-1
```

針對 Windows：

```
aws docdb describe-pending-maintenance-actions ^  
  --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^  
  --region us-east-1
```

此操作的輸出將會如下所示。

```
{  
  "PendingMaintenanceActions": [  
    {  
      "ResourceIdentifier": "arn:aws:rds:us-  
east-1:123456789012:cluster:sample-cluster",  
      "PendingMaintenanceActionDetails": [  
        {  
          "Description": "New feature",  
          "Action": "db-upgrade",  
          "ForcedApplyDate": "2019-02-25T21:46:00Z",  
          "AutoAppliedAfterDate": "2019-02-25T07:41:00Z",  
          "CurrentApplyDate": "2019-02-25T07:41:00Z"  
        }  
      ]  
    }  
  ]  
}
```

如果您的叢集有待定的維護，請繼續[對叢集的引擎版本執行修補程式更新](#)中的步驟。

對叢集的引擎版本執行修補程式更新

在本節中，我們將說明如何使用 AWS Management Console 或部署修補程式更新 AWS CLI。修補程式更新是相同引擎版本中的更新 (例如，將 3.6 引擎版本更新為較新的 3.6 引擎版本)。您可以立即更新，也可以在叢集的下一個維護時段進行更新。若要判斷引擎是否需要更新，請參閱[判斷擱置的維護](#)。請注意，當您套用更新時，叢集會經歷一些停機時間。

Note

如果您嘗試從主要引擎版本升級到另一個版本，例如 3.6 升級至 5.0，請參閱[Amazon DocumentDB 就地主要版本升級](#)或[使用升級您的 Amazon DocumentDB 集群 AWS Database Migration Service](#)。就地主要版本升級僅支援 docdb 5.0 作為目標引擎版本。

取得叢集引擎版本的最新修補程式更新需求有兩種組態需求：

- 叢集狀態必須為「可用」。
- 叢集必須執行舊版的引擎。

Using the AWS Management Console

下列程序會使用主控台將修補程式更新套用至叢集的引擎版本。您可以選擇立即更新，或在叢集的下一個維護時段期間進行更新。

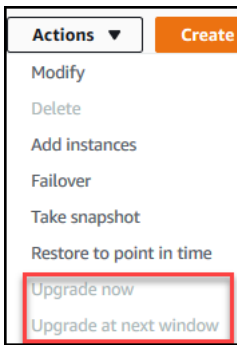
1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。在叢集清單中，選擇您想要升級之叢集的左側按鈕。叢集的狀態必須為「可用」。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示

(☰)

3. 從 Actions (動作) 功能表選擇下列其中一個選項。您選擇的叢集必須不是執行最新的引擎版本，這些功能表選項才可供選取。



- 立即升級 — 立即啟動升級程序。您的叢集會離線一段時間，讓叢集升級到最新的引擎版本。
 - 在下一個視窗升級 — 在叢集的下一個維護時段期間啟動升級程序。您的叢集在升級到最新的引擎版本時，會離線一段時間。
4. 當確認視窗開啟時，請選擇下列其中一項：
- 升級 — 根據上一步中選擇的排程，將叢集升級至最新的引擎版本。
 - 取消 — 取消叢集的引擎升級，並繼續叢集目前的引擎版本。

Using the AWS CLI

您可以使用 AWS CLI 和 `apply-pending-maintenance-action` 作業搭配下列參數，將修補程式更新套用至叢集。

參數

- **`--resource-identifier`** - 必要項目。您要升級的 Amazon DocumentDB 集群的 ARN。
- **`--apply-action`** - 必要項目。允許下列值。請使用 `db-upgrade` 升級叢集的引擎版本。
 - **`db-upgrade`**
 - **`system-update`**
- **`--opt-in-type`** - 必要項目。允許下列值。
 - `immediate` 立即套用維護動作。
 - `next-maintenance` 在下一個維護時段中套用維護動作。
 - `undo-opt-in` — 取消任何現有的 `next-maintenance` 選擇加入請求。

Example

下列範例修補程式 `sample-cluster` 將的引擎版本更新為 4.0.0 版。

若為 Linux、macOS 或 Unix：

```
aws docdb apply-pending-maintenance-action \  
  --resource-identifier arn:aws:rds:us-east-1:123456789012\:cluster:sample-cluster \  
 \  
  --apply-action db-upgrade \  
  --opt-in-type immediate
```

針對 Windows：

```
aws docdb apply-pending-maintenance-action ^ \  
  --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^ \  
  --apply-action db-upgrade ^ \  
  --opt-in-type immediate
```

此操作的輸出如下所示：

```
{  
  "ResourcePendingMaintenanceActions": {  
    "ResourceIdentifier": "arn:aws:rds:us-east-1:444455556666:cluster:docdb-2019-01-09-23-55-38",  
    "PendingMaintenanceActionDetails": [  
      {  
        "CurrentApplyDate": "2019-02-20T20:57:06.904Z",  
        "Description": "Bug fixes",  
        "ForcedApplyDate": "2019-02-25T21:46:00Z",  
        "OptInStatus": "immediate",  
        "Action": "db-upgrade",  
        "AutoAppliedAfterDate": "2019-02-25T07:41:00Z"  
      }  
    ]  
  }  
}
```

停止和啟動 Amazon DocumentDB 叢群

停止和啟動 Amazon DocumentDB 叢集可協助您管理開發和測試環境的成本。您不必在每次使用 Amazon DocumentDB 時建立和刪除叢集和執行個體，而是在不需要時暫時停止叢集中的所有執行個體。然後，您可以在恢復測試時再次啟動它們。

主題

- [停用和啟動叢集的概觀](#)
- [您可以在已停止的叢集上執行的作業](#)

停用和啟動叢集的概觀

在不需要 Amazon DocumentDB 叢集的期間，您可以一次停止該叢集中的所有執行個體。一旦您需要叢集，即可隨時重新啟動它。啟動和停用可簡化用於下列操作之叢集的設定和卸除程序：開發、測試或不需要連續可用性的類似活動。無論叢集中 AWS CLI 有多少個執行個體，您都可以使用 AWS Management Console 或以單一動作停止和啟動叢集。

雖然您的叢集已停止，但叢集儲存磁碟區保持不變。您只需支付指定的保留時段內儲存、手動快照和自動備份儲存的費用。您不需支付任何執行個體小時的費用。Amazon DocumentDB 會在七天後自動啟動叢集，因此不會落後於任何必要的維護更新之後。當您的叢集在七天後啟動時，您將再度開始對叢集中的執行個體付費。當您的叢集停止時，您無法查詢儲存磁碟區，因為查詢時需要執行個體處於可用狀態。

當 Amazon DocumentDB 叢集停止時，叢集和其執行個體都不能以任何方式修改。這包括新增或移除執行個體、或刪除叢集。

Using the AWS Management Console

下列程序說明如何停止有一或多個執行個體處於可用狀態的叢集，或停止叢集。

若要停止或啟動 Amazon DocumentDB 叢集

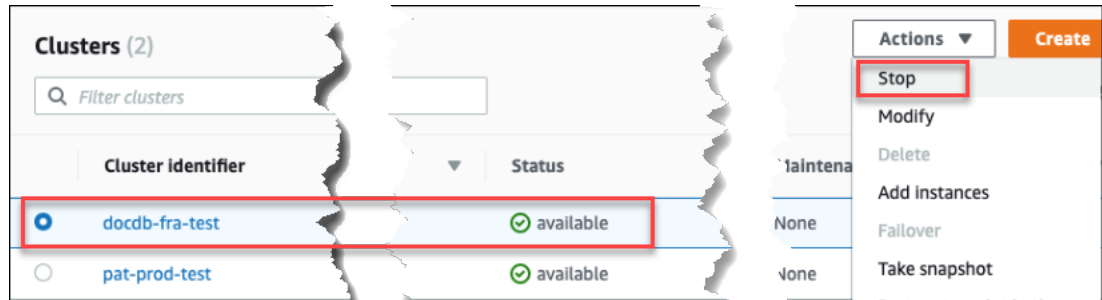
1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

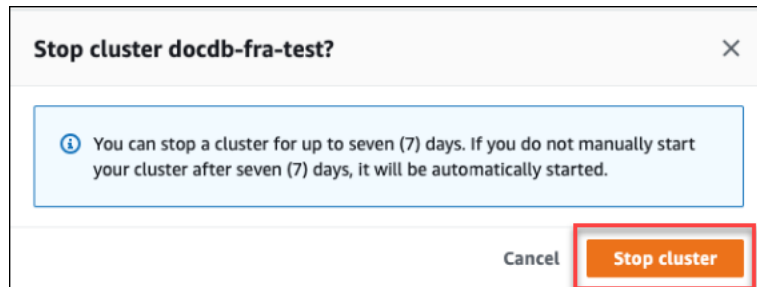
3. 在叢集清單中，選擇您要停止或啟動之叢集名稱左側的按鈕。
4. 選擇 Actions (動作)，然後選擇您想要在叢集上執行的動作。
 - 如果您想要停止可用的叢集：

a. 選擇停止。

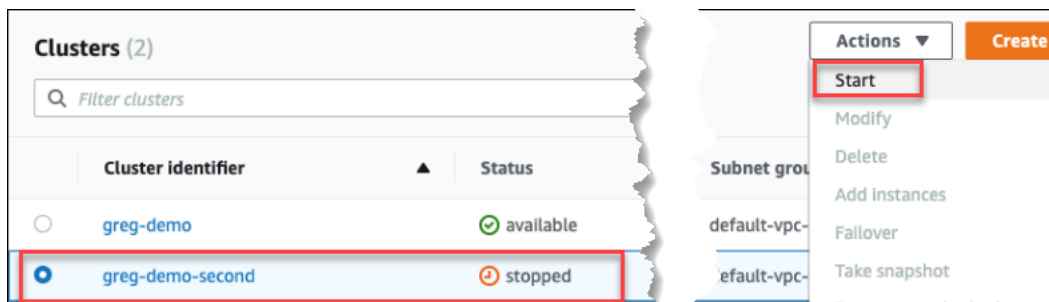
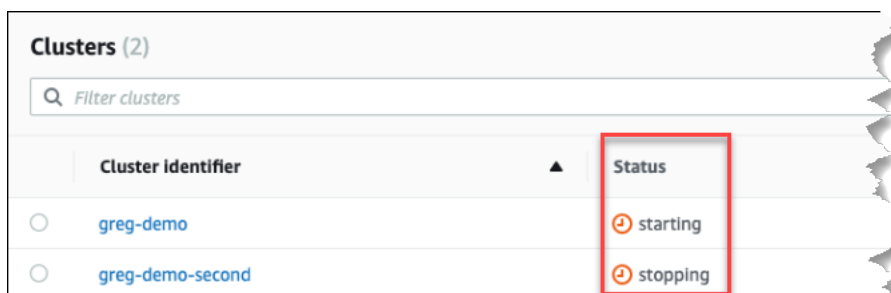


為了避免啟動容錯移轉機制，停止操作會先停止複本執行個體，然後停止主要執行個體。

b. 在確認對話方塊上，選擇 Stop cluster (停止叢集)，確認您要停止叢集，若要讓叢集持續執行，請選擇 Cancel (取消)。



• 如果您想要啟動叢集，而叢集已停止，請選擇 Start (啟動)。

5. 監控叢集及其執行個體的狀態。如果您已啟動叢集，當叢集及其執行個體均可用時，您即可恢復使用叢集。如需詳細資訊，請參閱 [決定叢集的狀態](#)。

Using the AWS CLI

下列程式碼範例說明如何停止有一或多個執行個體處於可用狀態的叢集，或停止叢集。

若要使用停止具有有一或多個可用執行個體的叢集 AWS CLI，請使用 `stop-db-cluster` 作業。若要啟動已停止的叢集，請使用 `start-db-cluster` 操作。這兩個操作均使用 `--db-cluster-identifier` 參數。

參數：

- **`--db-cluster-identifier`** - 必要項目。要停止或啟動的叢集名稱。

Example — 若要停止叢集，請使用 AWS CLI

下列程式碼會停止叢集 `sample-cluster`。叢集必須有一或多個處於可用狀態的執行個體。

若為 Linux、macOS 或 Unix：

```
aws docdb stop-db-cluster \  
  --db-cluster-identifier sample-cluster
```

針對 Windows：

```
aws docdb stop-db-cluster ^  
  --db-cluster-identifier sample-cluster
```

Example — 若要使用啟動叢集 AWS CLI

下列程式碼會啟動叢集 `sample-cluster`。叢集目前必須已停止。

若為 Linux、macOS 或 Unix：

```
aws docdb start-db-cluster \  
  --db-cluster-identifier sample-cluster
```

針對 Windows：

```
aws docdb start-db-cluster ^  
  --db-cluster-identifier sample-cluster
```

您可以在已停止的叢集上執行的作業

當 Amazon DocumentDB 叢集停止時，您可以將 point-in-time 還原到指定的自動備份保留期間內的任何時間點。如需執行 point-in-time 還原的詳細資訊，請參閱[時間點還原](#)。

叢集停止時，您無法修改 Amazon DocumentDB 叢集或其任何執行個體的組態。您也無法新增或移除叢集中的執行個體，或如果叢集仍有任何相關聯的執行個體，則無法刪除此叢集。您必須先啟動叢集，然後才能執行任何這類管理動作。

Amazon DocumentDB 只會在您已停止的叢集再次啟動之後，才會將任何排定的維護套用至該叢集。在七天之後，Amazon DocumentDB 會自動啟動已停止的叢集，這樣它就不會落後於其維護狀態太遠。當叢集重新啟動時，您將再度開始對叢集中的執行個體付費。

叢集停止時，Amazon DocumentDB 不會執行任何自動備份，也不會延長備份保留期限。

刪除 Amazon DocumentDB 集群

您可以使用 AWS Management Console 或刪除 Amazon DocumentDB 叢集。AWS CLI 若要刪除叢集，叢集必須處於可用狀態，且不得有任何與叢集相關聯的執行個體。如果叢集已停止，請先啟動叢集，等待叢集變成可用，然後刪除叢集。如需詳細資訊，請參閱[停止和啟動 Amazon DocumentDB 集群](#)。

刪除保護

為了避免您的叢集遭到意外刪除，您可以啟用刪除保護。當您使用主控台建立叢集時，預設會啟用刪除保護。不過，如果您使用 AWS CLI 建立叢集，預設會停用刪除保護。

無論您是使用控制台還是使用控制台執行刪除操作，Amazon DocumentDB 都會對叢集強制執行刪除保護。AWS CLI 如果已啟用刪除保護，您就無法刪除叢集。若要刪除已啟用刪除保護的叢集，您必須先修改叢集並停用刪除保護。

使用叢集已啟用刪除保護的主控台時，您就無法刪除叢集的最後一個執行個體，因為這麼做也會刪除叢集。您可以使用 AWS CLI 來刪除受刪除保護叢集的最後一個執行個體。不過，叢集本身仍會存在，而您的資料會保留下來。您可為叢集建立新的執行個體，以便存取資料。如需啟用和停用刪除保護的詳細資訊，請參閱：

- [創建一個 Amazon DocumentDB 集群](#)
- [修改 Amazon DocumentDB 叢集](#)

Using the AWS Management Console

若要使用刪除叢集 AWS Management Console，必須停用刪除保護。

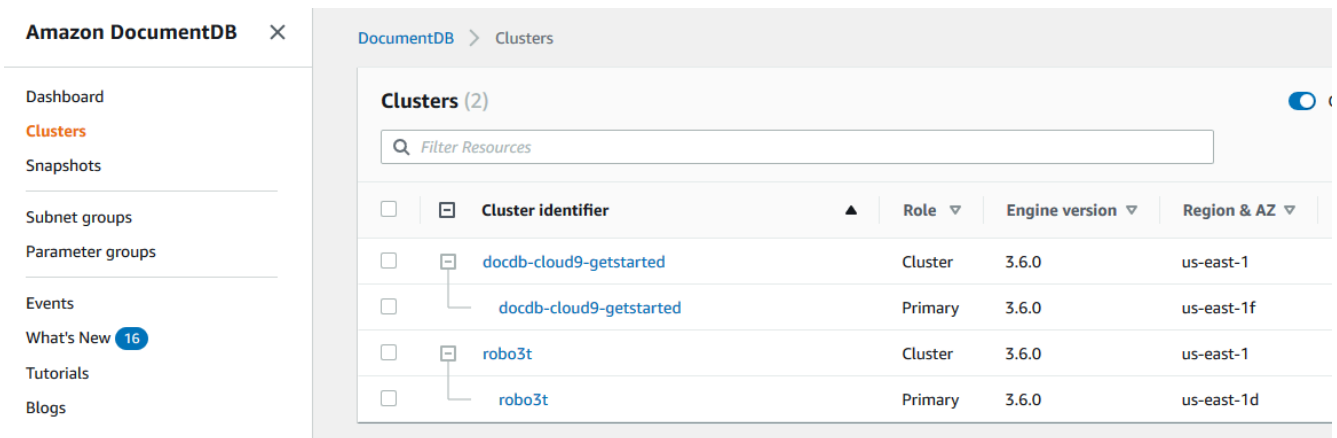
判斷叢集是否已啟用刪除保護：

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 請注意，在「叢集」導覽方塊中，「叢集識別碼」欄會同時顯示叢集和執行個體。執行個體會列在叢集下方，類似下方的螢幕擷取畫面。



4. 選擇叢集的名稱，然後選取 Configuration (組態) 索引標籤。在 Cluster details (叢集詳細資訊) 區段中，找出 Deletion protection (刪除保護)。如果已啟用刪除保護，請修改叢集以停用刪除保護。如需修改叢集的詳細資訊，請參閱[修改 Amazon DocumentDB 叢集](#)。

停用 Deletion protection (刪除保護) 之後，您就可以刪除叢集。

刪除叢集：

1. 在導覽窗格中，選擇叢集。
2. 檢查 Instances (執行個體) 欄，判斷叢集是否有任何執行個體。您必須先刪除叢集的所有執行個體，才能刪除叢集。如需詳細資訊，請參閱[刪除 Amazon DocumentDB 實例](#)。

3. 根據您的叢集是否有任何執行個體，執行下列其中一個步驟。
 - 如果叢集沒有執行個體，請選取叢集名稱左側的按鈕，然後選擇 Actions (動作)。從下拉式選單中選擇 Delete (刪除)。完成 Delete <cluster-name> (刪除 <cluster-name>) 對話方塊，然後選擇 Delete (刪除)。
 - 如果叢集有一或多個執行個體，請執行下列動作：
 - a. 在導覽窗格中，選擇執行個體。
 - b. 刪除叢集的每個執行個體。當您刪除最後一個執行個體時，叢集也會隨之刪除。如需刪除執行個體的相關資訊，請參閱[刪除 Amazon DocumentDB 實例](#)。

刪除叢集需要幾分鐘的時間。若要監控叢集的狀態，請參閱[監控 Amazon DocumentDB 集群的狀態](#)。

Using the AWS CLI

您無法刪除與任何執行個體相關聯的叢集。若要判斷哪些執行個體與您的叢集相關聯，請執行 `describe-db-clusters` 命令並刪除叢集的所有執行個體。然後視需要停用叢集上的刪除保護，最後再刪除叢集。

1. 首先，刪除叢集的所有執行個體。

若要判斷需要刪除哪些執行個體，請執行下列命令。

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].
  [DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[
  [
    "sample-cluster",
    [
      "sample-instance-1",
      "sample-instance-2"
    ]
  ]
]
```


如果您要刪除的叢集中有任何執行個體，請刪除它們，如下所示。

```
aws docdb delete-db-instance \  
  --db-instance-identifier sample-instance
```

2. 第二，停用刪除保護。

使用刪 AWS CLI 除叢集的所有執行個體並不會刪除叢集。您也必須刪除叢集，但只有已停用刪除保護，您才能這麼做。

若要判斷叢集是否已啟用刪除保護，請執行下列命令。

Tip

若要查看所有 Amazon DocumentDB 叢集的刪除保護狀態，請省略此 `--db-cluster-identifier` 參數。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[DBClusterIdentifier,DeletionProtection]'
```

此操作的輸出將會如下所示。

```
[  
  [  
    "sample-cluster",  
    "true"  
  ]  
]
```

如果已啟用刪除保護，請修改叢集並停用刪除保護。若要在叢集上停用刪除保護，請執行下列命令。

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --no-deletion-protection \  
  --apply-immediately
```

3. 最後，刪除叢集。

停用刪除保護之後，您就可以刪除叢集。若要刪除叢集，請使用 `delete-db-cluster` 操作搭配下列參數。

- **--db-cluster-identifier** - 必要項目。您要刪除之叢集的識別碼。
- **--final-db-snapshot-identifier** 選擇性。如果您要建立最終快照，則必須包含此參數及最終快照的名稱。您必須包含 `--final-db-snapshot-identifier` 或 `--skip-final-snapshot`。

命名限制：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- 對於跨 Amazon RDS、亞馬遜 Amazon Neptune 和 Amazon DocumentDB 的所有叢集，每 AWS 帳戶個區域的所有叢集都必須是唯一的。
- **--skip-final-snapshot** 選擇性。若您刪除叢集前不要建立最後的快照，才使用此參數。預設設定是建立最後的快照。您必須包含 `--final-db-snapshot-identifier` 或 `--skip-final-snapshot`。

下列 AWS CLI 程式碼會刪除 `sample-cluster` 具有最終快照的叢集。如有任何與叢集相關聯的執行個體，或已啟用刪除保護，操作就會失敗。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --final-db-snapshot-identifier sample-cluster-final-snapshot
```

針對 Windows：

```
aws docdb delete-db-cluster ^ \  
  --db-cluster-identifier sample-cluster ^
```

```
--final-db-snapshot-identifier sample-cluster-final-snapshot
```

Example

下面的 AWS CLI 代碼刪除集群，`sample-cluster`而不採取最終的快照。

若為 Linux、macOS 或 Unix：

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --skip-final-snapshot
```

針對 Windows：

```
aws docdb delete-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --skip-final-snapshot
```

`delete-db-cluster` 操作的輸出會是您要刪除的叢集。

刪除叢集需要幾分鐘的時間。若要監控叢集的狀態，請參閱[監視叢集的狀態](#)。

擴展 Amazon DocumentDB 集群

Amazon DocumentDB 可讓您根據自己的需求擴展叢集中的儲存和運算。本節說明如何使用儲存擴展、執行個體擴展和讀取擴展來管理 Amazon DocumentDB 叢集和執行個體的效能和擴展。

主題

- [儲存體擴展](#)
- [執行個體擴展](#)
- [讀取擴展](#)
- [寫入縮放](#)

儲存體擴展

Amazon DocumentDB 儲存會隨叢集磁碟區中的資料自動擴展。隨著資料的增長，叢集磁碟區儲存會以 10 GiB 的增量增長，最高可達 128 TiB。

執行個體擴展

您可以透過修改叢集中每個執行個體的執行個體類別，視需要擴展 Amazon DocumentDB 叢集。Amazon DocumentDB 支持針對 Amazon DocumentDB 优化的多個實例類別。

如需詳細資訊，請參閱 [修改 Amazon DocumentDB 實例](#)。

讀取擴展

您可以在叢集中建立最多 15 個 Amazon DocumentDB 複本，以實現 Amazon DocumentDB 叢集的讀取擴展。每個 Amazon DocumentDB 複本都會從叢集磁碟區傳回相同的資料，但最少的複本延遲時間 — 通常在主執行個體寫入更新後不到 100 毫秒。隨著讀取流量的增加，您可以建立額外的 Amazon DocumentDB 複本並直接連線到這些複本，以分配叢集的讀取負載。Amazon DocumentDB 複本不一定要與主執行個體屬於相同的執行個體類別。

如需詳細資訊，請參閱 [將 Amazon DocumentDB 執行個體新增至叢集](#)。

若要使用 Amazon DocumentDB 讀取擴展，建議您以複本集的形式連接到叢集，並使用驅動程式的內建讀取偏好設定功能將讀取分發到複本執行個體。如需詳細資訊，請參閱 [以複本集形式連接至 Amazon DocumentDB](#)


寫入縮放

您可以透過增加叢集主執行個體的大小來擴展 Amazon DocumentDB 叢集的寫入容量。本節提供兩種方法，可根據您的需求調整叢集的主要執行個體。第一個選項旨在將應用程式的影響降至最低，但需要更多步驟才能完成。第二個選項針對簡易性進行最佳化，因為它的步驟較少，但會對您的應用程式產生更多潛在的折衷影響。

視您的應用程式而定，您可以選擇最適合您的以下方法。如需有關可用執行個體大小和成本的詳細資訊，請參閱 [Amazon DocumentDB 定價頁面](#)。

1. 針對高可用性和效能進行最佳化 — 如果您要以 [複本集模式](#) 連線到叢集 (建議使用)，您可以使用下列程序將擴展主執行個體時對應用程式的影響降到最低。此方法可將影響降至最低，因為它使叢集維持在高可用性或高於高可用性的水準，而讀取擴展目標會作為執行個體新增至叢集，而不是就地更新。
 - a. 將一或多個較大執行個體類型的複本新增至叢集 (請參閱 [???](#))。建議所有複本具有與主要執行個體類型相同或更大的執行個體類型。這樣可以避免不小心降低寫入效能，從容錯移轉到較小的執行個體類型。對大多數客戶而言，這表示將叢集中的執行個體數目暫時增加一倍，然後在擴展完成後移除較小的複本。

- b. 將所有新複本上的容錯移轉層設定為優先順序為零，確保較小執行個體類型的複本具有最高的容錯移轉優先順序。如需詳細資訊，請參閱 [???](#)。
- c. 初始化手動容錯移轉，將其中一個新複本升級為主要執行個體。如需詳細資訊，請參閱 [???](#)。

 Note

這將導致您的叢集停機約 30 秒。請據此進行相應計劃。

- d. 從叢集中移除小於新主要執行個體類型的所有複本。
- e. 將所有執行個體的容錯移轉層設定回相同的優先順序 (通常，這表示會將它們設定回 1)。


舉例來說，假設您的叢集目前包含三個 r5.large 執行個體 (一個主要和兩個複本)，而您想要擴展到 r5.xlarge 執行個體類型。若要執行這項操作，您必須先將三個 r5.xlarge 複本執行個體新增至叢集，然後將新 r5.xlarge 複本的容錯移轉層設定為零。接下來，您將啟動手動容錯移轉 (您的應用程式將停機約 30 秒)。容錯移轉完成後，您會從叢集中移除所有三個 r5.large 執行個體，使叢集擴展至 r5.xlarge 執行個體。

為協助優化成本，Amazon DocumentDB 執行個體以一秒遞增方式計費，在建立、修改或刪除執行個體等可計費狀態變更後，最低收費十分鐘。如需詳細資訊，請參閱最佳實務文件中的 [成本最佳化](#)。

2. 優化簡單性-此方法優化簡單性。它不會擴展和收縮叢集，但可能會暫時降低您的讀取容量。

變更複本的執行個體類別可能會導致該執行個體在短暫的時間內無法提供要求，從幾秒鐘到少於 30 秒。如果您以 [複本集模式](#) 連線到叢集 (建議使用)，這會在擴充作業期間減少一個複本的讀取容量 (例如，3 節點叢集中的 66% 容量，或 4 節點叢集中 75% 的容量等)。

- a. 擴展叢集中的其中一個複本執行個體。如需詳細資訊，請參閱 [管理執行個體類別](#)。
- b. 請等待執行個體可用 (請參閱 [監控 Amazon DocumentDB 執行個體的狀態](#))。

 Note

這將導致您的叢集停機約 30 秒。請據此進行相應計劃。

- c. 繼續執行步驟 1 和 2，直到所有複本執行個體都已逐個調整為止。
- d. 起始手動容錯移轉。這會將其中一個複本提升為主要執行個體。如需詳細資訊，請參閱 [Amazon DocumentDB 容錯移轉](#)。

Note

這將導致叢集最多 30 秒的停機時間，但通常所花費的時間比這更少。請據此進行相應計劃。

- e. 擴展前一個主要執行個體 (現在是複本) 執行個體。

複製 Amazon DocumentDB 叢集的磁碟區

透過使用 Amazon DocumentDB 複製，您可以建立使用相同 Amazon DocumentDB 叢集磁碟區並具有與原始叢集相同資料的新叢集。該程序旨在快速且具有成本效益。與其相關聯的資料磁碟區新叢集稱為複製。與使用不同的技術 (如還原快照) 以物理方式複製資料相比，建立複製更快也更節省空間。

Amazon DocumentDB 支持從佈建的 Amazon DocumentDB 集群創建 Amazon DocumentDB 佈建的克隆。當您使用與來源不同的部署組態建立複製時，會使用最新版本的來源 Amazon DocumentDB 引擎建立複製。

當您從 Amazon DocumentDB 叢集建立翻製時，AWS 會在您的帳戶中建立複製，也就是擁有來源 Amazon DocumentDB 叢集的同帳戶。

主題

- [Amazon DocumentDB 克隆概述](#)
- [Amazon DocumentDB 克隆的局限性](#)
- [Amazon DocumentDB 克隆如何工作](#)
- [創建一個 Amazon DocumentDB 克隆](#)

Amazon DocumentDB 克隆概述

Amazon DocumentDB 使用 copy-on-write 協議來創建一個克隆。此機制會使用最少的其他空間來建立初始複製。首次建立複製時，Amazon DocumentDB 會保留來源資料庫叢集和新 (複製的) Amazon DocumentDB 叢集所使用之資料的單一副本。只有當來源 Amazon DocumentDB 叢集或 Amazon DocumentDB 叢集複製對資料 (在 Amazon DocumentDB 儲存磁碟區上) 進行變更時，才會配置額外的儲存體。若要進一步瞭解 copy-on-write 通訊協定，請參閱 [Amazon DocumentDB 克隆如何工作](#)。

對於使用生產資料快速設定測試環境而不會造成資料損壞風險，Amazon DocumentDB 複製特別有用。您可以為許多類型的應用程式使用複製，例如以下類型：

- 潛在變更實驗 (例如結構描述變化和參數群組變化)，以評估所有影響。
- 執行工作負載密集的操作，例如在複製上匯出資料或執行分析查詢。
- 為了開發、測試或其他用途建立生產資料庫叢集的副本。

您可以從相同的 Amazon DocumentDB 叢集建立多個複製。您也可以從另一個複製建立多個複製。

在建立 Amazon 文件資料庫複製之後，您可以將 Amazon DocumentDB 執行個體設定為不同於來源 Amazon DocumentDB 叢集的執行個體。例如，您可能不需要為了開發目的而複製，即可滿足與來源生產 Amazon DocumentDB 叢集相同的高可用性需求。在這種情況下，您可以使用單一 Amazon DocumentDB 執行個體來設定複製，而不是 Amazon DocumentDB 叢集使用的多個資料庫執行個體。

在您完成使用複製來進行測試、開發或其他用途時，便可將其刪除。

Amazon DocumentDB 克隆的局限性

Amazon DocumentDB; 複製目前有以下限制：

- 您可以根據需要建立數量不拘的複製項，最多可達到 AWS 區域中允許的資料庫叢集數量上限。但在建立 15 個複製項後，下一個複製項是完整複本。複製作業的作用類似於 point-in-time 復原。
- 您無法在來源 Amazon DocumentDB 叢集的不同 AWS 區域中建立複製。
- 您無法從沒有資料庫執行個體的 Amazon DocumentDB 叢集建立複製。您只能複製至少具有一個資料庫執行個體的 Amazon DocumentDB 叢集。
- 您可以在與 Amazon DocumentDB 叢集不同的虛擬私有雲端 (VPC) 中建立複本。不過，VPC 中的子網路必須映射至相同的可用區域。

Amazon DocumentDB 克隆如何工作

Amazon DocumentDB 複製工作在 Amazon DocumentDB 集群的存儲層。就支援 Amazon DocumentDB 儲存磁碟區的基礎耐久性媒體而言，它使用既快速又節省空間的 copy-on-write 通訊協定。您可以在中進一步了 Amazon DocumentDB 叢集磁碟區。[管理 Amazon DocumentDB 集群](#)

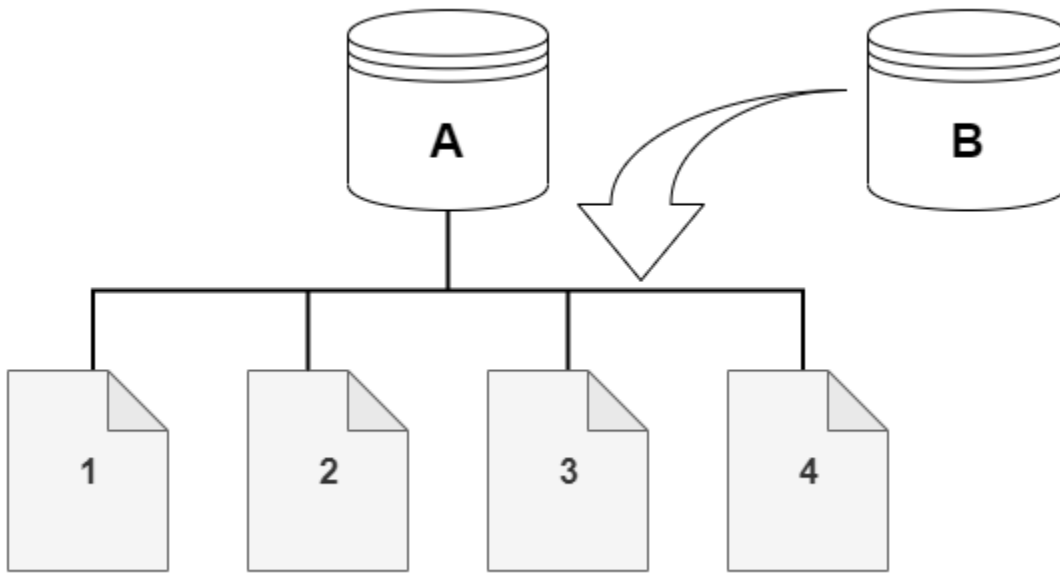
主題

- [了解 copy-on-write 協議](#)
- [刪除來源叢集磁碟區](#)

了解 copy-on-write 協議

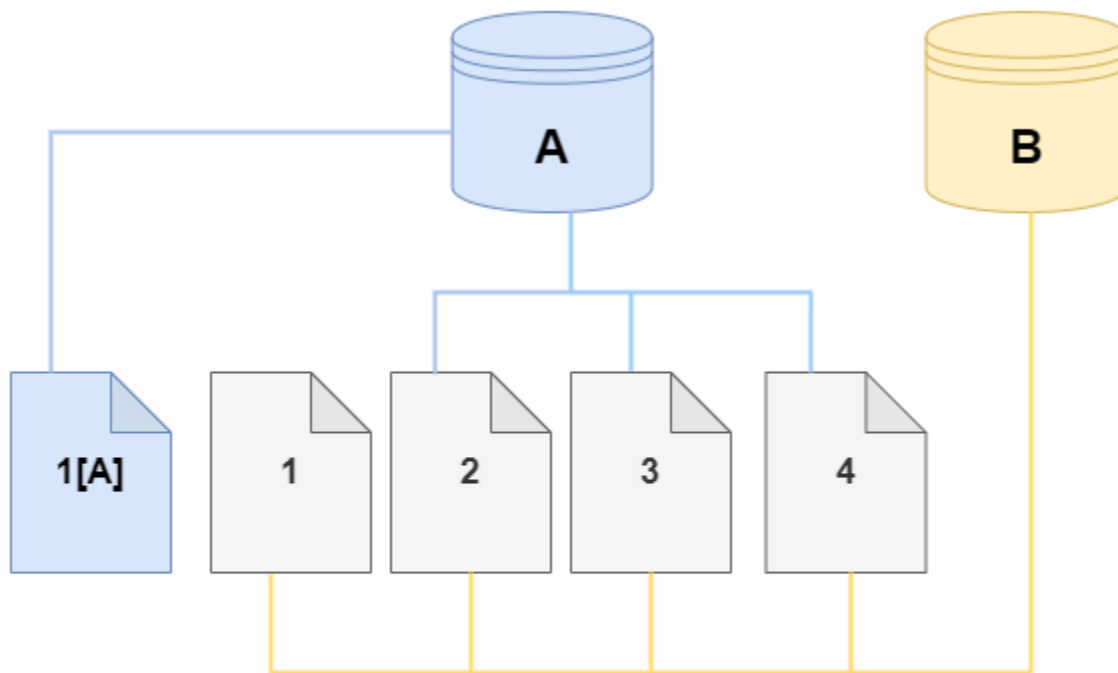
Amazon 文件資料庫叢集會將資料存放在基礎 Amazon DocumentDB 儲存磁碟區的頁面中。

例如，在下圖中，您可以找到具有四個資料頁 1、2、3 和 4 的 Amazon DocumentDB 叢集 (A)。想像一下，複製 B，是從 Amazon DocumentDB 集群創建的。建立複製時，不會複製任何資料。相反地，複製會指向與來源 Amazon DocumentDB 叢集相同的頁面集。

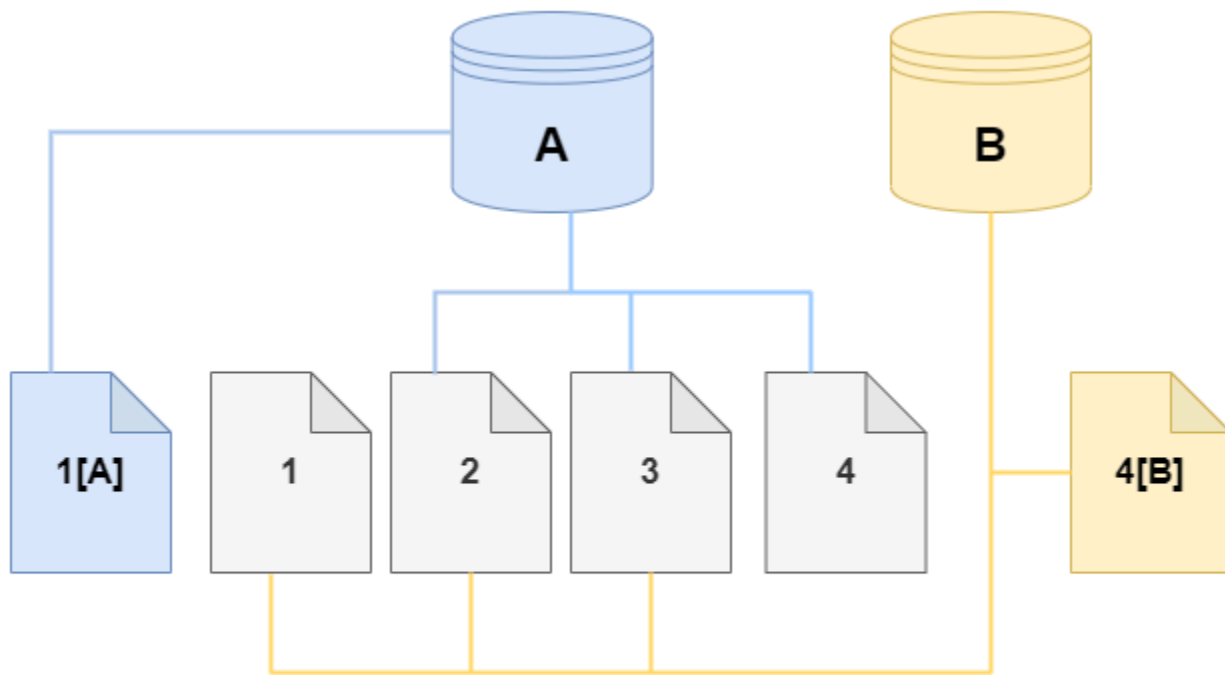


在建立複製時，通常不需要其他的儲存空間。copy-on-write 通訊協定在實體儲存媒體上使用與來源區段相同的區段。只有在來源區段的容量不足以容納整個複製區段時，才需要其他的儲存空間。如果是這種情況，來源區段會複製到另一個實體裝置上。

在下面的圖表中，你可以找到使用相同的集群 A 和它的克隆，B，如前面所示的行動 copy-on-write 協議的一個例子。假設您對 Amazon DocumentDB 叢集 (A) 進行了變更，這會導致頁面 1 上保留的資料發生變更。而不是寫入到原始頁面 1，Amazon DocumentDB 創建一個新的頁面 1 [A]。用於叢集 (A) 的 Amazon DocumentDB 叢集磁碟區現在指向第 1 頁 [A]、2、3 和 4，而複製 (B) 仍然參考原始頁面。



至於複製上，已對儲存磁碟區上的第 4 頁進行變更。而不是寫入到原始頁面 4，Amazon DocumentDB 創建一個新的頁面，4 [B]。複製現在會指向第 1、2、3 和 4 頁 [B]，而叢集 (A) 會繼續指向 1 [A]、2、3 和 4。



隨著來源 Amazon DocumentDB 叢集磁碟區和複製隨著時間的推移發生更多變更，因此擷取和儲存變更需要更多儲存空間。

刪除來源叢集磁碟區

在您刪除具有一個或多個與其相關聯複製的來源叢集磁碟區時，複製不會受到影響。複製會繼續指向叢集磁碟區先前所擁有的頁面。

創建一個 Amazon DocumentDB 克隆

您可以在與來源 Amazon DocumentDB 叢集相同的 AWS 帳戶中建立複製。若要執行此操作，您可以使用 AWS Management Console 或 AWS CLI 和下列程序。

透過使用 Amazon DocumentDB 複製，您可以從佈建的 Amazon DocumentDB 叢集建立佈建的 Amazon DocumentDB 叢集複製。

Using the AWS Management Console

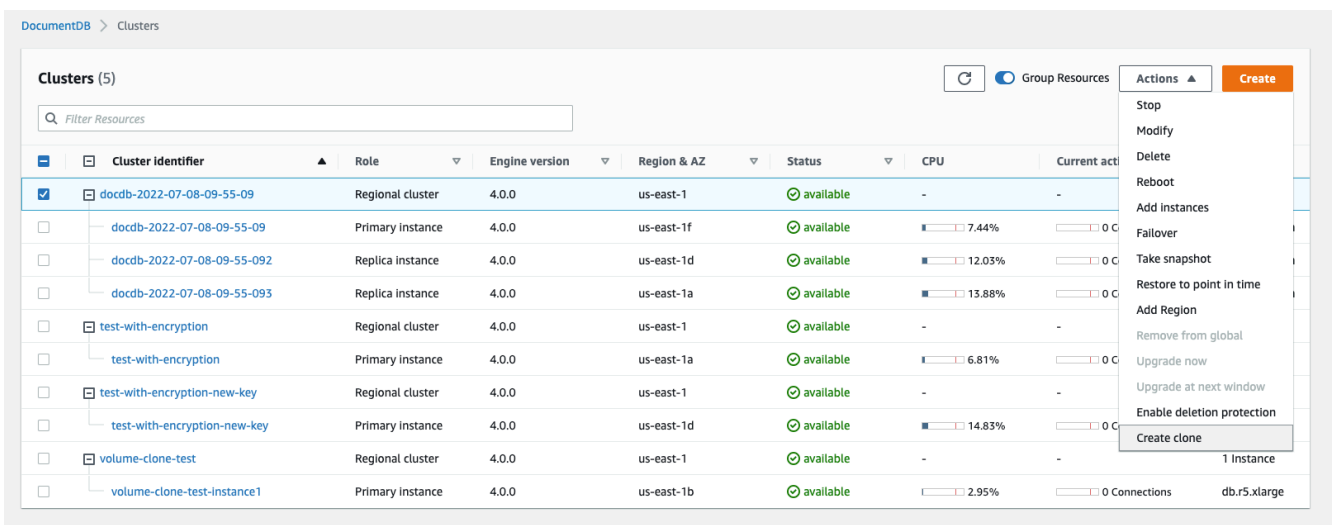
下列程序說明如何使用複製 Amazon DocumentDB 叢集。 AWS Management Console

使用具有一個 Amazon DocumentDB 執行個體的 Amazon DocumentDB 叢集中的 AWS Management Console 結果來建立複製。

這些指示適用於建立複製的相同 AWS 帳戶所擁有的資料庫叢集。資料庫叢集必須擁有相同的 AWS 帳戶，因為 Amazon DocumentDB 中不支援跨帳戶複製。

若要建立您 AWS 帳戶所擁有之資料庫叢集的複製，請使用 AWS Management Console

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。
3. 從清單中選擇您的 Amazon DocumentDB 叢集，然後針對動作選擇建立複製。



「建立複製」頁面隨即開啟，您可以在其中設定叢集識別碼和執行個體類別，以及 Amazon DocumentDB 叢集複製的其他選項。

4. 在 Settings (設定) 區段中，執行下列動作：
 - a. 對於叢集識別碼，請輸入您要提供給複製的 Amazon DocumentDB 叢集的名稱。
 - b. 對於執行個體組態，請為複製的 Amazon DocumentDB 叢集選取適當的執行個體類別。

Create Clone

You are cloning a DocumentDB cluster. This will create a new DB cluster that includes all of the data from the existing database as well as a writer DB instance.

Settings

Source cluster identifier

docdb-2022-07-08-09-55-09

Cluster identifier

Specify a unique cluster identifier.

new-cluster-clone

Instance configuration

Instance class

db.r6g.large

2 vCPUs 16GiB RAM

- c. 對於網路設定，請針對您的使用案例和相關的 VPC 安全性群組選擇子網路群組。
- d. 對於 Encryption-at-rest，如果來源叢集 (正在複製的叢集) 已啟用加密，則複製的叢集也必須啟用加密。如果此案例為 true，則 [啟用加密] 選項會變成灰色 (停用)，但已選取 [啟用加密] 選項。相反地，如果來源叢集未啟用加密，則可使用 [啟用加密] 選項，您可以選擇啟用或停用加密。

Network settings

Subnet group
A subnet group is a collection of subnets that are within a VPC.

default ▼

VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups ▼

default ✕

Encryption-at-rest

Enable encryption

Enable encryption
 Disable encryption

KMS key ID

(default) aws/rds ▼

Account
12345678910

KMS key ID
example-key-abcdef123

- e. 選取要匯出的記錄檔類型 (選擇性)、輸入用於連線至叢集的特定連接埠，以及啟用不小心刪除叢集的防護 (預設為啟用)，以完成新的叢集複製配置。

Log exports

Select the log types to publish to Amazon CloudWatch Logs

Audit logs

Profiler logs

Cluster options

Port
TCP/IP port that is used to connect to the cluster.

27017

Deletion protection

Enable deletion protection
Protects the cluster from being accidentally deleted. While this option is enabled, you can't delete the cluster.

Tags

No tags associated with the cluster.

Add new tag

You can add 50 more tags.

Cancel **Create**

- f. 完成輸入 Amazon DocumentDB 叢集複製的所有設定。若要進一步了解 Amazon DocumentDB 叢集和執行個體設定，請參閱。[管理 Amazon DocumentDB 集群](#)
5. 選擇建立複製以啟動您選擇的 Amazon DocumentDB 叢集的 Amazon DocumentDB 複製。

複製建立時，複製會與您的其他 Amazon DocumentDB 叢集一起列在主控台資料庫區段中，並顯示其目前狀態。在您複製的狀態為 Available (可用) 時，表示已可使用。

Using the AWS CLI

使用 AWS CLI 複製您的 Amazon DocumentDB 叢集需要執行幾個步驟。

您使用的 `restore-db-cluster-to-point-in-time` AWS CLI 命令會產生一個空的 Amazon DocumentDB 叢集，其中包含 0 個 Amazon DocumentDB 執行個體。也就是說，該命令只還原 Amazon DocumentDB 叢集，而不會還原該叢集的資料庫執行個體。您可以在複製可用之後分別執行該動作。程序珠的兩個步驟如下：

1. 使用 [restore-db-cluster-to-point-in-time](#) CLI 指令建立翻製。您與此命令搭配使用的參數可控制所建立之空白 Amazon DocumentDB 叢集 (複製) 的容量類型和其他詳細資料。
2. 使用 [create-db-instance](#) CLI 命令在還原的 Amazon DocumentDB 叢集中重新建立 Amazon DocumentDB 執行個體，為複製建立 Amazon DocumentDB 執行個體。

以下指令假設 AWS CLI 已將「AWS 地區」設定為預設值。此方法可以避免在每個密令中傳遞 `--region` 名稱。如需詳細資訊，請參閱[設定 AWS CLI](#)。您還可以指定後續每個 CLI 命令中的 `--region`。

建立翻製

您傳遞給 [restore-db-cluster-to-point-in-time](#) CLI 命令的特定參數皆有所不同。傳遞的內容取決於您要建立的翻製類型。

使用下列程序，從佈建的 Amazon DocumentDB 叢集建立佈建的 Amazon DocumentDB 複製。

若要建立與來源 Amazon DocumentDB 叢集相同引擎模式的複製

- 使用 [restore-db-cluster-to-point-in-time](#) CLI 命令，並指定下列參數的值：
 - `--db-cluster-identifier` – 為您的複製選擇一個有意義的名稱。您可以在使用 [restore-db-cluster-to-point-in-time](#) CLI 指令時命名翻製。
 - `--restore-type` – 使用 `copy-on-write` 來建立來源資料庫叢集的複製。如果沒有這個參數，就會 `restore-db-cluster-to-point-in-time` 還原 Amazon DocumentDB 叢集，而不是建立複製。預設值 `restore-type` 為 `full-copy`。
 - `--source-db-cluster-identifier` – 使用您要複製的來源 Amazon DocumentDB 叢集的名稱。
 - `--use-latest-restorable-time` – 此值會指向複製的最新可還原磁碟區資料。此參數是必需的 `restore-type copy-on-write`，但是，您不能 `restore-to-time parameter` 與它一起使用。

下列範例會從名為 `my-source-cluster` 的叢集建立名為 `my-clone` 的複製。

針對 Linux、macOS 或 Unix：

```
aws docdb restore-db-cluster-to-point-in-time \  
  --source-db-cluster-identifier my-source-cluster \  
  --db-cluster-identifier my-clone \  
  --restore-type copy-on-write \  
  --use-latest-restorable-time
```

針對 Windows：

```
aws docdb restore-db-cluster-to-point-in-time ^  
  --source-db-cluster-identifier my-source-cluster ^  
  --db-cluster-identifier my-clone ^  
  --restore-type copy-on-write ^  
  --use-latest-restorable-time
```

該命令會返回包含複製詳細資訊的 JSON 對象。在嘗試為複製建立資料庫執行個體之前，請檢查以確定您複製的資料庫叢集是否可用。如需詳細資訊，請參閱下方的檢查狀態和取得複製詳細資料：

檢查狀態並取得複製詳細資料

您可以使用下列命令，來檢查您新建立之空白資料庫叢集的狀態。

```
$ aws docdb describe-db-clusters --db-cluster-identifier my-clone --query '*[].[Status]' --output text
```

或者，您可以使用下列 AWS CLI 查詢取得為複製建立資料庫執行個體所需的狀態和其他值：

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone \  
  --query '*[].[Status:Status,Engine:Engine,EngineVersion:EngineVersion]'
```

針對 Windows：

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone ^  
  --query '*[].[Status:Status,Engine:Engine,EngineVersion:EngineVersion]'
```


此查詢會返回類似以下的內容：

```
[
  {
    "Status": "available",
    "Engine": "docdb",
    "EngineVersion": "4.0.0",
  }
]
```

為您的複製建立 Amazon DocumentDB 執行個體

使用 [create-db-instance](#) CLI 命令為複製建立資料庫執行個體。

此 `--db-instance-class` 參數僅用於佈建的 Amazon DocumentDB 叢集。

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-instance \
  --db-instance-identifier my-new-db \
  --db-cluster-identifier my-clone \
  --db-instance-class db.r5.4xlarge \
  --engine docdb
```

針對 Windows：

```
aws docdb create-db-instance ^
  --db-instance-identifier my-new-db ^
  --db-cluster-identifier my-clone ^
  --db-instance-class db.r5.4xlarge ^
  --engine docdb
```

用於複製的參數

下表摘要說明與複製 Amazon DocumentDB 叢集搭配 `restore-db-cluster-to-point-in-time` 使用的各種參數。

| 參數 | 描述 |
|---|------------------------------------|
| <code>--source-db-cluster-identifier</code> | 使用您要複製的來源 Amazon DocumentDB 叢集的名稱。 |

| 參數 | 描述 |
|---|--|
| <code>--db-cluster-identifier</code> | 為您的複製選擇一個有意義的名稱。您使用 <code>restore-db-cluster-to-point-in-time</code> 命令為您的複製命名。然後，您將此名稱傳遞給 <code>create-db-instance</code> 命令。 |
| <code>--restore-type</code> | 指定 <code>copy-on-write</code> 為 <code>--restore-type</code> 以建立來源資料庫叢集的複本，而不是還原來源 Amazon DocumentDB 叢集。 |
| <code>--use-latest-restorable-time</code> | 此值會指向複製的最新可還原磁碟區資料。 |

了解 Amazon DocumentDB 叢集容錯

Amazon DocumentDB 叢集在設計上具有容錯能力。每個叢集的磁碟區會跨越單一的多個可用區域 AWS 區域，而每個可用區域都包含叢集磁碟區資料的複本。此功能意味著叢集可承受可用區域故障，完全不會遺失資料，服務只會短暫中斷。

如果叢集中的主要執行個體發生故障，Amazon DocumentDB 會以下列兩種方式之一自動執行容錯移轉到新的主執行個體：

- 藉由將現有 Amazon DocumentDB 複本提升至根據每個複本的促銷層設定選擇的新主要執行個體，然後建立先前主要執行個體的取代方案。複本執行個體的容錯移轉通常只需不到 30 秒。在此期間，讀取和寫入作業可能會遭遇短暫的中斷。若要提高叢集的可用性，建議您在兩個或多個不同的可用區域中建立至少一個或多個 Amazon DocumentDB 複本。
- 建立新的主要執行個體。只有當您的叢集中沒有複本執行個體，而且可能需要幾分鐘的時間才會發生這種情況。

如果叢集具有一或多個 Amazon DocumentDB 複本，則 Amazon DocumentDB 複本會在發生故障事件期間提升為主執行個體。失敗事件會導致短暫中斷，在此期間，讀取和寫入操作會失敗，並引發例外狀況。不過，服務通常會在 120 秒之內恢復，往往不超過 60 秒。若要提高叢集的可用性，建議您在兩個或多個不同的可用區域中建立至少一個或多個 Amazon DocumentDB 複本。

您可以透過為每個複本指派優先順序，自訂 Amazon DocumentDB 複本在發生故障後升級到主執行個體的順序。優先順序從 0 (代表最高優先順序) 到 15 (代表最低優先順序)。如果主執行個體發生故障，具有最高優先順序的 Amazon DocumentDB 複本會提升為新的主執行個體。您可以隨時修改 Amazon DocumentDB 複本的優先順序。修改優先順序不會觸發容錯移轉。您可以搭配 `--promotion-tier`

參數使用 `modify-db-instance` 操作。如需更多有關自訂執行個體容錯移轉優先順序的詳細資訊，請參閱[Amazon DocumentDB 容錯移轉](#)。

多個 Amazon DocumentDB 複本可以共用相同的優先順序，進而產生促銷等級。如果兩個或多個 Amazon DocumentDB 複本具有相同的優先順序，則大小最大的複本會提升為主要複本。如果兩個或多個 Amazon DocumentDB 複本共用相同的優先順序和大小，則會提升相同促銷方案中的任意複本。

如果叢集不包含任何 Amazon DocumentDB 複本，則會在發生故障事件期間重新建立主執行個體。失敗事件會導致中斷，在此期間，讀取和寫入操作會失敗，並引發例外狀況。建立新的主要執行個體後，服務就會恢復，通常不超過 10 分鐘。將 Amazon DocumentDB 複本升級到主執行個體比建立新的主執行個體快得多。

管理 Amazon DocumentDB 實例

下列主題提供的資訊可協助您管理 Amazon DocumentDB 執行個體。當中包括有關執行個體類別和狀態，以及如何建立、刪除和修改執行個體的詳細資訊。

主題

- [管理執行個體類別](#)
- [決定執行個體的狀態](#)
- [Amazon DocumentDB 實例生命週](#)

管理執行個體類別

執行個體類別會決定 Amazon DocumentDB (具有 MongoDB 相容性) 執行個體的運算和記憶體容量。您需要的執行個體類別取決於您的處理能力和記憶體需求。

Amazon DocumentDB 支援執行個體類別的 R4、R5、R6G、T3 和 T4G 系列。這些類別是新一代的執行個體類別，已針對記憶體密集型應用程式進行最佳化處理。如需這些類別的規格，請參閱[執行個體類別規格](#)。

主題

- [決定執行個體類別](#)
- [變更執行個體的類別](#)
- [各區域的支援執行個體類別](#)
- [執行個體類別規格](#)

決定執行個體類別

若要判斷執行個體的類別，您可以使用 AWS Management Console 或 `describe-db-instances` AWS CLI 作業。

Using the AWS Management Console

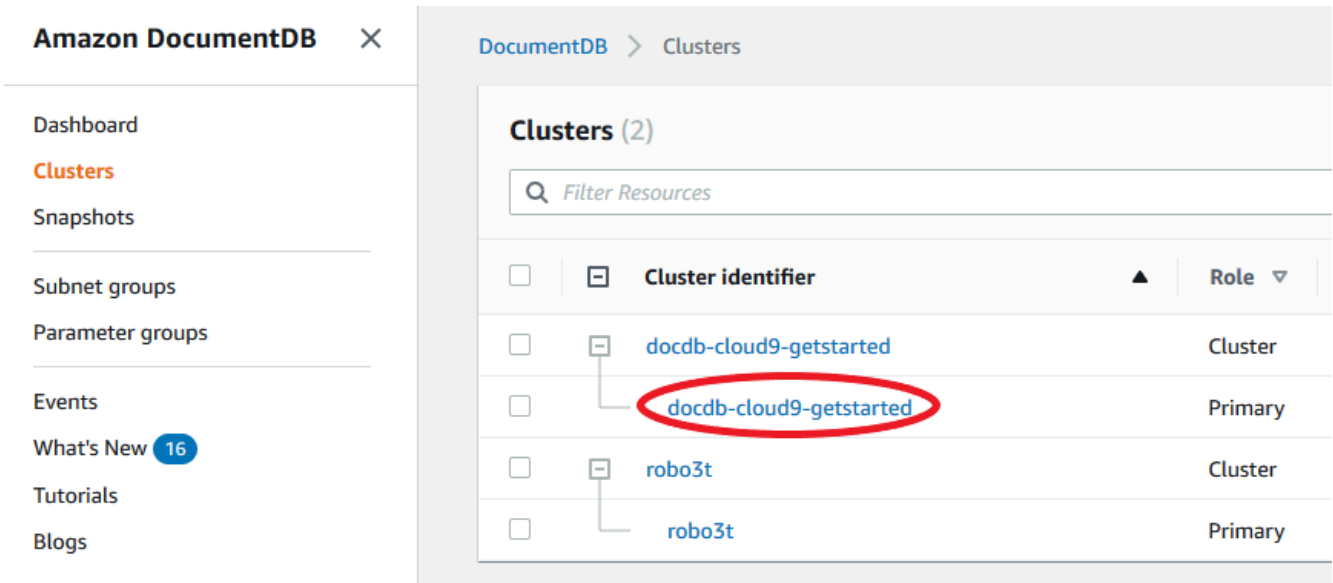
若要判斷叢集執行個體的執行個體類別，請在主控台中完成下列步驟。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在瀏覽窗格中，選擇 [叢集] 以尋找您感興趣的執行個體。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 [叢集] 導覽方塊中，您會看到 [叢集識別碼] 資料行。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。



The screenshot shows the Amazon DocumentDB console interface. On the left is a navigation sidebar with options like Dashboard, Clusters, Snapshots, Subnet groups, Parameter groups, Events, What's New (16), Tutorials, and Blogs. The main content area is titled 'DocumentDB > Clusters' and displays a table of clusters. The table has columns for checkboxes, cluster identifiers, and roles. One cluster, 'docdb-cloud9-getstarted', is expanded to show its primary instance, which is circled in red. Another cluster, 'robo3t', is also expanded to show its primary instance.

| | Cluster identifier | Role |
|--------------------------|-------------------------|---------|
| <input type="checkbox"/> | docdb-cloud9-getstarted | Cluster |
| <input type="checkbox"/> | docdb-cloud9-getstarted | Primary |
| <input type="checkbox"/> | robo3t | Cluster |
| <input type="checkbox"/> | robo3t | Primary |

4. 在執行個體清單中，展開叢集以尋找您感興趣的執行個體。尋找您想要的執行個體。接著，查看執行個體資料列的「大小」欄，查看其實體類別。

在下圖中，執行個體 `robo3t` 的執行個體類別為 `db.r5.4xlarge`。

DocumentDB > Clusters

Clusters (2) Group Resources Actions Create

Filter Resources

| Cluster identifier | Role | Engine version | Region & AZ | Status | Size | Maintenance |
|-------------------------|---------|----------------|-------------|-----------|-------------|-------------|
| docdb-cloud9-getstarted | Cluster | 3.6.0 | us-east-1 | available | 1 Instance | None |
| docdb-cloud9-getstarted | Primary | 3.6.0 | us-east-1f | available | db.r5.large | None |
| robo3t | Cluster | 3.6.0 | us-east-1 | available | 1 Instance | None |
| robo3t | Primary | 3.6.0 | us-east-1d | available | db.r5.large | None |

Using the AWS CLI

若要使用判斷執行個體的類別 AWS CLI，請搭配下列參數使用此 `describe-db-instances` 作業。

- **--db-instance-identifier**-可選。指定您要尋找其執行個體類別的執行個體。如果省略這個參數，`describe-db-instances` 會傳回最多 100 個執行個體的敘述。
- **--query**-可選。指定要納入結果中的執行個體成員。如果省略此參數，則所有執行個體成員都會傳回。

Example

下列範例會尋找執行個體的執行個體名稱和類別 `sample-instance-1`。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
  --db-instance-identifier sample-instance-1
```

針對 Windows：

```
aws docdb describe-db-instances ^
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
  --db-instance-identifier sample-instance-1
```

此操作的輸出將會如下所示。

```
[
  [
    "sample-instance-1",
    "db.r5.large"
  ]
]
```

Example

下列範例會尋找最多 100 個 Amazon DocumentDB 執行個體的執行個體名稱和類別。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
  --filter Name=engine,Values=docdb
```

針對 Windows：

```
aws docdb describe-db-instances ^
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
  --filter Name=engine,Values=docdb
```

此操作的輸出將會如下所示。

```
[
  [
    "sample-instance-1",
    "db.r5.large"
  ],
  [
    "sample-instance-2",
    "db.r5.large"
  ],
  [
    "sample-instance-3",
    "db.r5.4xlarge"
  ],
  [
    "sample-instance-4",
    "db.r5.4xlarge"
  ]
]
```

如需詳細資訊，請參閱 [描述 Amazon DocumentDB 實例](#)。

變更執行個體的類別

您可以使用 AWS Management Console 或來變更執行個體的執行個體類別 AWS CLI。如需詳細資訊，請參閱 [修改 Amazon DocumentDB 實例](#)。

各區域的支援執行個體類別

Amazon DocumentDB 支持以下實例類：

- R6G— 最新一代的記憶體最佳化執行個體，採用 ARM 式 AWS Graviton2 處理器，比 R5 執行個體提供高達 30% 的效能，成本便宜 5%。
- R5— 記憶體最佳化執行個體，以相同的執行個體成本提供比 R4 執行個體高達 100% 的效能。
- R4 上一代記憶體最佳化例證。
- T4G— 最新一代的低成本高載通用執行個體類型，由 ARM 式 AWS Graviton2 處理器提供支援，可提供基準等級的 CPU 效能，比 T3 執行個體提供高達 35% 的價格效能，非常適合執行 CPU 使用率適中的應用程式，而且使用暫時性尖峰的應用程式。
- T3— 低成本的高載通用執行個體類型，可提供 CPU 效能的基準等級，並可在需要的時間內隨時突增 CPU 使用率。

如需這些執行個體類別的詳細規格，請參閱 [執行個體類別規格](#)。

一個特定的實例類可能會或可能不支持在給定的區域。下表指定 Amazon DocumentDB 在每個區域支援哪些執行個體類別。

各區域的支援執行個體類別

| 區域 | R6G | R5 | R4 | T4G | T3 |
|---------------|-----|----|----|-----|----|
| 美國東部 (俄亥俄) | 支援 | 支援 | 支援 | 支援 | 支援 |
| 美國東部 (維吉尼亞北部) | 支援 | 支援 | 支援 | 支援 | 支援 |
| 美國西部 (奧勒岡) | 支援 | 支援 | 支援 | 支援 | 支援 |

| 區域 | R6G | R5 | R4 | T4G | T3 |
|----------------|-----|----|----|-----|----|
| 南美洲 (聖保羅) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (香港) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (海德拉巴) | | 支援 | | | 支援 |
| 亞太區域 (孟買) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (首爾) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (悉尼) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (新加坡) | 支援 | 支援 | | 支援 | 支援 |
| 亞太區域 (東京) | 支援 | 支援 | | 支援 | 支援 |
| 加拿大 (中部) | 支援 | 支援 | | 支援 | 支援 |
| 歐洲 (法蘭克福) | 支援 | 支援 | | 支援 | 支援 |
| 歐洲 (愛爾蘭) | 支援 | 支援 | 支援 | 支援 | 支援 |
| 歐洲 (倫敦) | 支援 | 支援 | | 支援 | 支援 |
| 歐洲 (米蘭) | 支援 | 支援 | | 支援 | 支援 |
| Europe (Paris) | 支援 | 支援 | | 支援 | 支援 |

| 區域 | R6G | R5 | R4 | T4G | T3 |
|---------------------|-----|----|----|-----|----|
| 中東 (阿拉伯聯合大公國) | 支援 | 支援 | | 支援 | 支援 |
| 中國 (北京) 區域 | 支援 | 支援 | | 支援 | 支援 |
| 中國 (寧夏) | 支援 | 支援 | | 支援 | 支援 |
| AWS GovCloud (美國西部) | 支援 | 支援 | | 支援 | 支援 |
| AWS GovCloud (美國東部) | 支援 | 支援 | | 支援 | 支援 |

執行個體類別規格

下表提供 Amazon 文件資料庫執行個體類別的詳細資訊。您可以在表的下方找到每個表格欄的說明。

支援的 Amazon DocumentDB 執行個體類別

| 執行個體類別 | vCPU ¹ | 記憶體 (GiB) ² | 最大儲存溫度 (GiB) ³ | 最大頻寬 (Mbps) ⁴ | 網路效能 ⁵ | 支援引擎 ⁶ |
|--------|-------------------|------------------------|---------------------------|--------------------------|-------------------|-------------------|
|--------|-------------------|------------------------|---------------------------|--------------------------|-------------------|-------------------|

R6G — 基於重力 2 的最新一代記憶體最佳化執行個體類別

| | | | | | | |
|------------------|---|----|-----|----------|------------|-------|
| db.r6g.1a.large | 2 | 16 | 32 | 最高 4,750 | 最高 10 Gbps | 四點零及五 |
| db.r6g.x1a.large | 4 | 32 | 63 | 最高 4,750 | 最高 10 Gbps | 四點零及五 |
| db.r6g.2xlarge | 8 | 64 | 126 | 最高 4,750 | 最高 10 Gbps | 四點零及五 |

| 執行個體類別 | vCPU ¹ | 記憶體 (GiB) ² | 最大儲存溫度 (GiB) ³ | 最大頻寬 (Mbps) ⁴ | 網路效能 ⁵ | 支援引擎 ⁶ |
|-----------------|-------------------|------------------------|---------------------------|--------------------------|-------------------|-------------------|
| db.r6g.4xlarge | 16 | 128 | 252 | 4,750 | 最高 10 Gbps | 四點零及五 |
| db.r6g.8xlarge | 32 | 256 | 504 | 9,000 | 12 Gbps | 四點零及五 |
| db.r6g.12xlarge | 48 | 384 | 756 | 13,500 | 20 Gbps | 四點零及五 |
| db.r6g.16xlarge | 64 | 512 | 1008 | 19,000 | 25 Gbps | 四點零及五 |

R5 — 上一代記憶體最佳化執行個體類別

| | | | | | | |
|----------------|----|-----|------|----------|------------|---------------|
| db.r5.large | 2 | 16 | 31 | 最高 3,500 | 最高 10 Gbps | 3.6.0、4.0.0 和 |
| db.r5.xlarge | 4 | 32 | 62 | 最高 3,500 | 最高 10 Gbps | 3.6.0、4.0.0 和 |
| db.r5.2xlarge | 8 | 64 | 124 | 最高 3,500 | 最高 10 Gbps | 3.6.0、4.0.0 和 |
| db.r5.4xlarge | 16 | 128 | 249 | 3,500 | 最高 10 Gbps | 3.6.0、4.0.0 和 |
| db.r5.8xlarge | 32 | 256 | 504 | 6,800 | 10 Gbps | 3.6.0、4.0.0 和 |
| db.r5.12xlarge | 48 | 384 | 748 | 7,000 | 10 Gbps | 3.6.0、4.0.0 和 |
| db.r5.16xlarge | 64 | 512 | 1008 | 13,600 | 20 Gbps | 3.6.0、4.0.0 和 |
| db.r5.24xlarge | 96 | 768 | 1500 | 14,000 | 25 Gbps | 3.6.0、4.0.0 和 |

| 執行個體類別 | vCPU ¹ | 記憶體 (GiB) ² | 最大儲存溫度 (GiB) ³ | 最大頻寬 (Mbps) ⁴ | 網路效能 ⁵ | 支援引擎 ⁶ |
|------------------------------------|-------------------|------------------------|---------------------------|--------------------------|-------------------|-------------------|
| R4 — 上一代記憶體最佳化執行個體類別 | | | | | | |
| db.r4.large | 2 | 15.25 | 30 | 437 | 最高 10 Gbps | 僅限 |
| db.r4.xlarge | 4 | 30.5 | 60 | 875 | 最高 10 Gbps | 僅限 |
| db.r4.2xlarge | 8 | 61 | 120 | 875 | 最高 10 Gbps | 僅限 |
| db.r4.4xlarge | 16 | 122 | 240 | 875 | 最高 10 Gbps | 僅限 |
| db.r4.8xlarge | 32 | 244 | 480 | 875 | 10 Gbps | 僅限 |
| db.r4.16xlarge | 64 | 488 | 960 | 14,000 | 25 Gbps | 僅限 |
| T4G — 基於重力的最新一代高載效能執行個體類別 2 | | | | | | |
| db.t4g.medium | 2 | 4 | 8.13 | 最高 2,085 | 最高 5 Gbps | 四點零及五 |
| T3 — 上一代高載效能執行個體類別 | | | | | | |
| db.t3.medium | 2 | 4 | 7.5 | 最高 1,536 | 最高 5 Gbps | 3.6.0、4.0 .0 和 |

| 執行個體類別 | vCPU ¹ | 記憶體 (GiB) 2 | 最大儲存溫度 (GiB) 3 | 最大頻寬 (Mbps) ⁴ | 網路效能 ⁵ | 支援引擎 ⁶ |
|--------|-------------------|-------------|----------------|--------------------------|-------------------|-------------------|
|--------|-------------------|-------------|----------------|--------------------------|-------------------|-------------------|

1. vCPU — 虛擬中央處理器 (CPU) 的數目。虛擬 CPU 即為容量單位，可用來比較執行個體類別。您不再購買或租用特定的處理器並使用數月或數年，而是以小時為單位租用容量。我們的目標是無論實際的基礎硬體為何，均提供一致的 CPU 容量。
2. 記憶體 (GiB) — 配置給執行個體的 RAM (以 GB 為單位)。記憶體和 vCPU 之間的比率通常是固定的。
3. 最大溫度儲存體 (GiB) — 為非持續性暫存檔案儲存而配置給執行個體的 RAM (以 GB 為單位)。
4. 最大頻寬 (Mbps) — 以每秒百萬位元為單位的最大頻寬。若將頻寬除以 8，即可取得預期的傳輸量 (MB/秒)。
5. 網路效能 — 相對於其他執行個體類別的網路速度。
6. 支援引擎 — 支援執行個體類別的 Amazon DocumentDB 料庫謎團。

決定執行個體的狀態

若要查看有效的執行個體狀態、它們的意義，以及如何判斷執行個體狀態的詳細資訊，請參閱[監控 Amazon DocumentDB 執行個體的狀態](#)。

Amazon DocumentDB 實例生命週

Amazon DocumentDB 執行個體的生命週期包括建立、修改、維護和升級、執行備份和還原、重新啟動和刪除執行個體。本節提供如何完成這些程序的相關資訊。

主題

- [將 Amazon DocumentDB 執行個體新增至叢集](#)
- [描述 Amazon DocumentDB 實例](#)
- [修改 Amazon DocumentDB 實例](#)
- [重新啟動 Amazon DocumentDB 實例](#)
- [刪除 Amazon DocumentDB 實例](#)

您可以使用 AWS Management Console 或建立新的 Amazon DocumentDB 執行個體。AWS CLI 若要將執行個體新增至叢集，叢集必須處於可用狀態。您無法將執行個體新增至已停止的叢集。如果叢集

已停止，請先啟動叢集，等待叢集變成可用，然後新增執行個體。如需詳細資訊，請參閱 [停止和啟動 Amazon DocumentDB 叢群](#)。

Note

如果您使用主控台建立 Amazon DocumentDB 叢集，系統會同時為您自動建立執行個體。如果您要建立其他執行個體，請使用下列程序之一。

將 Amazon DocumentDB 執行個體新增至叢集

Using the AWS Management Console

使用下列程序，使用 Amazon DocumentDB 主控台為叢集建立執行個體。

1. [登入 AWS Management Console](#)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 若要選擇要新增執行個體的叢集，請選取叢集名稱左側的按鈕。
4. 選擇 Actions (動作)，然後選擇 Add instances (新增執行個體)。
5. 在 Add instance to: (新增執行個體至)<cluster-name> 頁面，針對您想要加到叢集的每個執行個體重複以下步驟。您最多可以擁有 15 個。
 - a. 執行個體識別碼 — 您可以為此執行個體輸入唯一識別碼，或允許 Amazon DocumentDB 根據叢集識別碼提供執行個體識別碼。

執行個體命名限制：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

- Amazon RDS、Neptune 和 Amazon DocumentDB 中每 AWS 帳戶個區域的所有執行個體必須是唯一的。
 - b. 實例類別 — 從下拉式清單中選擇要用於此實例的實例類型。
 - c. 促銷層 — 從下拉式清單中選擇執行個體的促銷方案，或選擇 [無偏好設定] 以允許 Amazon DocumentDB 為您的執行個體設定促銷方案。數字越小表示優先順序越高。如需詳細資訊，請參閱 [控制容錯移轉目標](#)。
 - d. 若要新增更多執行個體，請選擇 Add additional instances (新增其他執行個體)，然後重複步驟 a、b 和 c。
6. 完成操作。
- 若要新增執行個體到您的叢集，請選擇 Create (建立)。
 - 若要取消操作，請選擇 Cancel (取消)。

建立執行個體需要幾分鐘才能完成。您可以使用主控台或 AWS CLI 檢視執行個體的状态。如需詳細資訊，請參閱 [監視執行個體的状态](#)。

Using the AWS CLI

使用 create-db-instance AWS CLI 作業搭配下列參數來建立叢集的主要執行個體。

- **--db-instance-class** – 必要。執行個體的運算與記憶體容量，例如 db.m4.large。並非所有實例類都可用 AWS 區域。
- **--db-instance-identifier** – 必要。識別執行個體的字串。

執行個體命名限制條件：

- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。
- Amazon RDS、Neptune 和 Amazon DocumentDB 中每 AWS 帳戶個區域的所有執行個體必須是唯一的。
- **--engine** – 必要。必須為 docdb。
- **--availability-zone**-可選。要在其中建立執行個體的可用區域。使用此參數尋找不同可用區域中的執行個體，以提高容錯能力。如需詳細資訊，請參閱 [Amazon DocumentDB 高可用性和複寫](#)。
- **--promotion-tier**-可選。此執行個體的容錯移轉優先順序方案。必須介於 0 和 15 之間，數字越小優先順序越高。如需詳細資訊，請參閱 [控制容錯移轉目標](#)。

1. 首先，決定您可以在其中建立執行個體的可用區域。

如果您想在建立執行個體之前指定可用區域，請執行下列命令來判斷 Amazon DocumentDB 叢集可用的可用區域。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \  
    --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

針對 Windows：

```
aws docdb describe-db-clusters ^\  
    --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

此操作的輸出將會如下所示。

```
[  
  [  
    "sample-cluster",  
    [  
      "us-east-1c",  
      "us-east-1b",  
      "us-east-1a"  
    ]  
  ]  
]
```

2. 其次，判定您可以在區域中建立哪些執行個體類別。

若要判斷您區域中可用的執行個體類別，請執行以下命令。從輸出中，為您要新增至 Amazon DocumentDB 叢集的執行個體選擇一個執行個體類別。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-orderable-db-instance-options \  
    --engine docdb \  
    --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

針對 Windows：

```
aws docdb describe-orderable-db-instance-options ^
  --engine docdb ^
  --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

此操作的輸出將會如下所示。

```
[
  "db.r5.16xlarge",
  "db.r5.2xlarge",
  "db.r5.4xlarge",
  "db.r5.8xlarge",
  "db.r5.large",
  "db.r5.xlarge"
]
```

3. 最後，將執行個體新增至您的 Amazon DocumentDB 叢集。

若要將執行個體新增至 Amazon DocumentDB 叢集，請執行下列命令。

若為 Linux、macOS 或 Unix：

```
aws docdb create-db-instance \
  --db-cluster-identifier sample-cluster \
  --db-instance-identifier sample-instance-2 \
  --availability-zone us-east-1b \
  --promotion-tier 2 \
  --db-instance-class db.r5.xlarge \
  --engine docdb
```

針對 Windows：

```
aws docdb create-db-instance ^
  --db-cluster-identifier sample-cluster ^
  --db-instance-identifier sample-instance-2 ^
  --availability-zone us-east-1b ^
  --promotion-tier 2 ^
  --db-instance-class db.r5.xlarge ^
  --engine docdb
```

此操作的輸出將會如下所示。


```
{
  "DBInstance": {
    "DBInstanceIdentifier": "sample-instance-2",
    "DBInstanceClass": "db.r5.xlarge",
    "Engine": "docdb",
    "DBInstanceStatus": "creating",
    "PreferredBackupWindow": "02:00-02:30",
    "BackupRetentionPeriod": 1,
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-abcd0123",
        "Status": "active"
      }
    ],
    "AvailabilityZone": "us-east-1b",
    "DBSubnetGroup": {
      "DBSubnetGroupName": "default",
      "DBSubnetGroupDescription": "default",
      "VpcId": "vpc-6242c31a",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-abcd0123",
          "SubnetAvailabilityZone": {
            "Name": "us-west-2a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-wxyz0123",
          "SubnetAvailabilityZone": {
            "Name": "us-west-2b"
          },
          "SubnetStatus": "Active"
        }
      ]
    },
    "PreferredMaintenanceWindow": "sun:11:35-sun:12:05",
    "PendingModifiedValues": {},
    "EngineVersion": "3.6.0",
    "AutoMinorVersionUpgrade": true,
    "PubliclyAccessible": false,
    "DBClusterIdentifier": "sample-cluster",
  }
}
```

```
"StorageEncrypted": true,  
"KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",  
"DbiResourceId": "db-ABCDEFGHIJKLMNQPQRSTUVWXYZ",  
"CACertificateIdentifier": "rds-ca-2019",  
"PromotionTier": 2,  
"DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2"  
}  
}
```

建立執行個體需要幾分鐘才能完成。您可以使用主控台或 AWS CLI 檢視執行個體的狀態。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 執行個體的狀態](#)。

描述 Amazon DocumentDB 實例

您可以使用 Amazon DocumentDB 管理主控台或查看與 Amazon DocumentDB 執行個體相關的連線端點、安全群組 VPC、憑證授權單位和參數群組等詳細資訊。AWS CLI

Using the AWS Management Console

若要使用 AWS Management Console 檢視執行個體的詳細資訊，請依照下列步驟操作。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示

(☰)

)。

3. 在 [叢集] 導覽方塊中，您會看到 [叢集識別碼] 資料行。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。

| <input type="checkbox"/> | <input type="checkbox"/> | Cluster identifier | Role |
|--------------------------|--------------------------|-------------------------|---------|
| <input type="checkbox"/> | <input type="checkbox"/> | docdb-cloud9-getstarted | Cluster |
| <input type="checkbox"/> | <input type="checkbox"/> | docdb-cloud9-getstarted | Primary |
| <input type="checkbox"/> | <input type="checkbox"/> | robo3t | Cluster |
| <input type="checkbox"/> | <input type="checkbox"/> | robo3t | Primary |

4. 在執行個體清單中，選擇您要查看其詳細資訊的執行個體名稱。執行個體的相關資訊分為下列群組：
- 摘要 — 執行個體的一般資訊，包括引擎版本、類別、狀態和任何擱置中的維護。
 - 連線與安全性 — 「Connect 線」區段會列出使用 mongo shell 或應用程式連線至此執行個體的連線端點。Security Groups (安全群組) 區段會列出與此執行個體相關聯的安全群組，以及其 VPC ID 和描述。
 - 組態 — 詳細資料區段列出執行個體的組態和狀態，包括執行個體的 Amazon 資源名稱 (ARN)、端點、角色、類別和憑證授權單位。它也會列出執行個體的安全性和網路設定，以及備份資訊。Cluster details (叢集詳細資訊) 區段會列出此執行個體所屬叢集的詳細資訊。Cluster instances (叢集執行個體) 區段會列出屬於您叢集的所有執行個體，當中每個執行個體的角色和叢集參數群組狀態。

Note

您可以選取叢集詳細資料標頭旁的 [修改]，修改與執行個體相關聯的叢集。如需詳細資訊，請參閱 [修改 Amazon DocumentDB 叢集](#)。

- 監控 — 此執行處理的 CloudWatch 記錄測量結果。如需詳細資訊，請參閱 [使用監控亞馬遜文檔數據庫 CloudWatch](#)。
- 事件與標籤 — 最近的事件區段會列出此執行個體的最近事件。Amazon DocumentDB 會保留與叢集、執行個體、快照、安全群組和叢集參數群組相關的事件記錄。此資訊包括與每個事件相關聯的日期、時間和訊息。Tags (標籤) 區段會列出連接至此叢集的標籤。如需詳細資訊，請參閱 [標記亞馬遜文檔數據庫資源](#)。

Using the AWS CLI

若要使用檢視 Amazon DocumentDB 執行個體的詳細資訊 AWS CLI，請使用下 `describe-db-clusters` 列範例所示的命令。如需詳細資訊，請參閱 Amazon DocumentDB 源管理 API 參考 [DescribeDBInstances](#) 中的。

Note

對於某些管理功能，例如叢集和執行個體生命週期管理，Amazon DocumentDB 會運用與 Amazon RDS 共用的操作技術。`filterName=engine,Values=docdb` 篩選器參數只會傳回 Amazon DocumentDB 叢集。

1. 列出所有 Amazon DocumentDB 實例。

下列 AWS CLI 程式碼列出區域中所有 Amazon DocumentDB 執行個體的詳細資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \  
  --filter Name=engine,Values=docdb
```

針對 Windows：

```
aws docdb describe-db-instances \  
  --filter Name=engine,Values=docdb
```

2. 列出指定 Amazon 文件資料庫執行個體的所有詳細資訊

以下程式碼會列出 `sample-cluster-instance` 的詳細資訊。將 `--db-instance-identifier` 參數加入執行個體名稱會將輸出限制為該特定執行個體的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \  
  --db-instance-identifier sample-cluster-instance
```

針對 Windows：

```
aws docdb describe-db-instances \  
  --db-instance-identifier sample-cluster-instance
```

```
--db-instance-identifier sample-cluster-instance
```

此操作的輸出如下所示：

```
{
  "DBInstances": [
    {
      "DbiResourceId": "db-BJKKB54PIDV5QFKGVRX5T3S6GM",
      "DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-
cluster-instance-00",
      "VpcSecurityGroups": [
        {
          "VpcSecurityGroupId": "sg-77186e0d",
          "Status": "active"
        }
      ],
      "DBInstanceClass": "db.r5.large",
      "DBInstanceStatus": "creating",
      "AutoMinorVersionUpgrade": true,
      "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",
      "BackupRetentionPeriod": 1,
      "StorageEncrypted": true,
      "DBClusterIdentifier": "sample-cluster",
      "EngineVersion": "3.6.0",
      "AvailabilityZone": "us-east-1a",
      "Engine": "docdb",
      "PromotionTier": 2,
      "DBInstanceIdentifier": "sample-cluster-instance",
      "PreferredBackupWindow": "00:00-00:30",
      "PubliclyAccessible": false,
      "DBSubnetGroup": {
        "DBSubnetGroupName": "default",
        "Subnets": [
          {
            "SubnetIdentifier": "subnet-4e26d263",
            "SubnetAvailabilityZone": {
              "Name": "us-east-1a"
            },
            "SubnetStatus": "Active"
          },
          {
            "SubnetIdentifier": "subnet-afc329f4",
            "SubnetAvailabilityZone": {
```

```
        "Name": "us-east-1c"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-b3806e8f",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1e"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-53ab3636",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1d"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-991cb8d0",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1b"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-29ab1025",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1f"
      },
      "SubnetStatus": "Active"
    }
  ],
  "VpcId": "vpc-91280df6",
  "DBSubnetGroupDescription": "default",
  "SubnetGroupStatus": "Complete"
},
"PendingModifiedValues": {},
"KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-
a50b-44d4-b6a0-a177d5ff730b"
}
]
```

修改 Amazon DocumentDB 實例

您可以使用 AWS Management Console 或修改 Amazon DocumentDB 執行個體。AWS CLI若要修改執行個體，執行個體必須處於可用狀態。您無法修改已停止的執行個體。如果叢集已停止，請先啟動叢集，等待執行個體變成可用，然後進行所需的修改。如需詳細資訊，請參閱 [停止和啟動 Amazon DocumentDB 叢群](#)。

Using the AWS Management Console

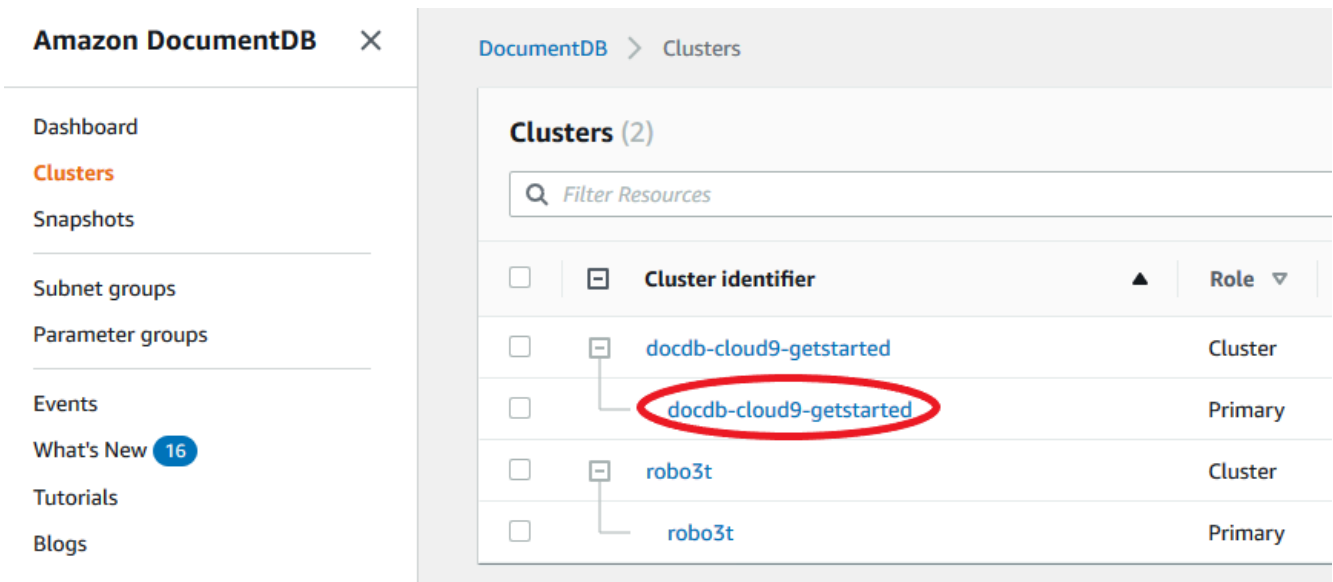
若要使用主控台修改特定的 Amazon DocumentDB 執行個體，請完成以下步驟。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 [叢集] 導覽方塊中，您會看到 [叢集識別碼] 資料行。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。



The screenshot shows the Amazon DocumentDB console interface. On the left is a navigation sidebar with options like Dashboard, Clusters, Snapshots, Subnet groups, Parameter groups, Events, What's New (16), Tutorials, and Blogs. The main content area is titled 'DocumentDB > Clusters' and displays a table of clusters. The table has columns for 'Cluster identifier' and 'Role'. The cluster 'docdb-cloud9-getstarted' is selected, and its primary instance is circled in red.

| Cluster identifier | Role |
|-------------------------|---------|
| docdb-cloud9-getstarted | Cluster |
| docdb-cloud9-getstarted | Primary |
| robo3t | Cluster |
| robo3t | Primary |

4. 核取您要修改之執行個體左側的方塊。

5. 選擇 Actions (動作)，然後選擇 Modify (修改)。
6. 在 Modify instance:<instance-name> (修改執行個體：<執行個體-名稱> 窗格中，進行您想要的變更。您可進行下列變更。
 - 執行個體規格 — 執行個體識別碼和類別。執行個體識別符命名限制：
 - 執行個體識別碼 — 輸入您 AWS 帳戶 在目前地區擁有的所有執行個體的唯一名稱。實體識別碼必須包含 [1—63] 英數字元或連字號，以字母作為第一個字元，且不能以連字號結尾或包含兩個連字號。
 - 執行個體類別 — 從下拉式功能表中，選取 Amazon DocumentDB 執行個體的執行個體類別。如需詳細資訊，請參閱 [管理執行個體類別](#)。
 - 憑證授權單位 — 此執行個體的伺服器憑證。如需詳細資訊，請參閱 [更新您的 Amazon DocumentDB TLS 證書](#)。
 - 容錯移轉 — 在容錯移轉期間，具有最高升級層的執行個體將升級為主要執行個體。如需詳細資訊，請參閱 [Amazon DocumentDB 容錯移轉](#)。
 - 維護 — 將擱置修改或修正程式套用至叢集中執行處理的維護時段。
7. 完成後，請選擇 Continue (繼續) 以查看變更的摘要。
8. 確認變更後，即可立即套用，或是在下一個維護時段在 Scheduling of modifications (修改的排程) 下套用。選擇 Modify Instance (修改執行個體)，以儲存變更。或者，您可以選擇 Cancel (取消) 以捨棄變更。

套用您的變更需要幾分鐘的時間才會完成。您只能使用狀態為可用的執行個體。您可以使用主控台或 AWS CLI 監控執行個體的状态。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 執行個體的状态](#)。

Using the AWS CLI

若要使用修改特定的 Amazon DocumentDB 執行個體 AWS CLI，請 `modify-db-instance` 搭配下列參數使用。如需詳細資訊，請參閱 [ModifyDBInstance](#)。以下程式碼將執行個體 `sample-instance` 的執行個體類別修改為 `db.r5.large`。

參數

- **--db-instance-identifier** – 必要。要修改的執行個體識別碼。
- **--db-instance-class** – 可選。執行個體的新運算和記憶體容量；例如，`db.r5.large`。並非所有實例類都可用 AWS 區域。如果您修改執行個體類別，變更期間就會發生中斷。除非針對此要求指定為 `true`，否則 `ApplyImmediately` 變更會在下一個維護時段套用。

- **--apply-immediately**或 **--no-apply-immediately**-可選。指定這項修改應立即套用，或等到下一次維護時段才套用。若省略此參數，則會在下一次維護時段進行修改。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb modify-db-instance \  
  --db-instance-identifier sample-instance \  
  --db-instance-class db.r5.large \  
  --apply-immediately
```

針對 Windows：

```
aws docdb modify-db-instance ^  
  --db-instance-identifier sample-instance ^  
  --db-instance-class db.r5.large ^  
  --apply-immediately
```

此操作的輸出將會如下所示。

```
{  
  "DBInstances": [  
    {  
      "DBInstanceIdentifier": "sample-instance-1",  
      "DBInstanceClass": "db.r5.large",  
      "Engine": "docdb",  
      "DBInstanceStatus": "modifying",  
      "Endpoint": {  
        "Address": "sample-instance-1.node.us-east-1.docdb.amazonaws.com",  
        "Port": 27017,  
        "HostedZoneId": "ABCDEFGHIJKLM"  
      },  
      "InstanceCreateTime": "2020-01-10T22:18:55.921Z",  
      "PreferredBackupWindow": "02:00-02:30",  
      "BackupRetentionPeriod": 1,  
      "VpcSecurityGroups": [  
        {  
          "VpcSecurityGroupId": "sg-abcd0123",  
          "Status": "active"  
        }  
      ],  
    },  
  ],  
}
```

```

    "AvailabilityZone": "us-east-1a",
    "DBSubnetGroup": {
      "DBSubnetGroupName": "default",
      "DBSubnetGroupDescription": "default",
      "VpcId": "vpc-abcd0123",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-abcd0123",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-abcd0123",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1b"
          },
          "SubnetStatus": "Active"
        }
      ]
    },
    "PreferredMaintenanceWindow": "sun:10:57-sun:11:27",
    "PendingModifiedValues": {
      "DBInstanceClass": "db.r5.large"
    },
    "EngineVersion": "3.6.0",
    "AutoMinorVersionUpgrade": true,
    "PubliclyAccessible": false,
    "DBClusterIdentifier": "sample-cluster",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/wJalrXUtnFEMI/
K7MDENG/bPxRfiCYEXAMPLEKEY",
    "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
    "CACertificateIdentifier": "rds-ca-2019",
    "PromotionTier": 1,
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:sample-
instance-1",
    "EnabledCloudwatchLogsExports": [
      "profiler"
    ]
  }
]

```

```
}
```

套用您的修改需要幾分鐘的時間才會完成。您只能使用狀態為可用的執行個體。您可以使用 AWS Management Console 或監視執行個體的状态 AWS CLI。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 執行個體的状态](#)。

重新啟動 Amazon DocumentDB 實例

有時候，您可能需要重新啟動 Amazon DocumentDB 執行個體，通常是出於維護原因。如果您進行某些變更，例如變更與叢集相關聯的叢集參數群組，則必須重新啟動叢集中的執行個體讓變更生效。您可以使用 AWS Management Console 或重新啟動指定的執行個體 AWS CLI。

重新啟動執行個體，將重新啟動資料庫引擎服務。重新啟動會導致暫時中斷，在此期間，執行個體狀態會設定為 `rebooting`。重新開機完成後，就會建立 Amazon DocumentDB 事件。

重新啟動執行個體不會導致容錯移轉。若要容錯移轉 Amazon DocumentDB 叢集，請使用 AWS Management Console 或作業。AWS CLI `failover-db-cluster` 如需詳細資訊，請參閱 [Amazon DocumentDB 容錯移轉](#)。

如果執行個體不是可用狀態，就無法重新啟動執行個體。資料庫無法使用可能由多種原因導致，例如，先前要求的修改，或維護時段的動作。如需執行個體状态的詳細資訊，請參閱 [監控 Amazon DocumentDB 執行個體的状态](#)。

Using the AWS Management Console

以下程序會重新啟動您使用主控台指定的執行個體。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 [叢集] 導覽方塊中，您會看到 [叢集識別碼] 資料行。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。

The screenshot shows the Amazon DocumentDB console interface. On the left is a navigation menu with options like Dashboard, Clusters, Snapshots, Subnet groups, Parameter groups, Events, What's New (16), Tutorials, and Blogs. The main content area is titled 'DocumentDB > Clusters' and shows a list of clusters under the heading 'Clusters (2)'. There is a search bar labeled 'Filter Resources'. The list has columns for checkboxes, cluster identifiers, and roles. The cluster 'docdb-cloud9-getstarted' is circled in red, and its role is 'Primary'. Other clusters include 'robo3t' with a 'Cluster' role and another 'robo3t' with a 'Primary' role.

| <input type="checkbox"/> | <input type="checkbox"/> | Cluster identifier | Role |
|--------------------------|--------------------------|-------------------------|---------|
| <input type="checkbox"/> | <input type="checkbox"/> | docdb-cloud9-getstarted | Cluster |
| <input type="checkbox"/> | <input type="checkbox"/> | docdb-cloud9-getstarted | Primary |
| <input type="checkbox"/> | <input type="checkbox"/> | robo3t | Cluster |
| <input type="checkbox"/> | <input type="checkbox"/> | robo3t | Primary |

4. 勾選您要重新啟動的執行個體左側的核取方塊。
5. 選擇 Actions (動作)、選擇 Reboot (重新啟動)，然後選擇 Reboot (重新啟動) 以確認重新啟動。

執行個體重新啟動需要幾分鐘的時間來完成。您只能使用狀態為可用的執行個體。您可以使用主控台或 AWS CLI 監控執行個體的状态。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 執行個體的状态](#)。

Using the AWS CLI

若要重新啟動 Amazon DocumentDB 執行個體，請搭配參數使用此 `reboot-db-instance` 作業。 `--db-instance-identifier` 此參數會指定要重新啟動之執行個體的識別符。

以下程式碼會重新啟動執行個體 `sample-instance`。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb reboot-db-instance \
  --db-instance-identifier sample-instance
```

針對 Windows：

```
aws docdb reboot-db-instance ^
  --db-instance-identifier sample-instance
```

此操作的輸出將會如下所示。

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "sample-instance",
    "DBInstanceClass": "db.r5.large",
    "Engine": "docdb",
    "DBInstanceStatus": "rebooting",
    "Endpoint": {
      "Address": "sample-instance.node.us-east-1.docdb.amazonaws.com",
      "Port": 27017,
      "HostedZoneId": "ABCDEFGHIJKLM"
    },
    "InstanceCreateTime": "2020-03-27T08:05:56.314Z",
    "PreferredBackupWindow": "02:00-02:30",
    "BackupRetentionPeriod": 1,
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-abcd0123",
        "Status": "active"
      }
    ],
    "AvailabilityZone": "us-east-1c",
    "DBSubnetGroup": {
      "DBSubnetGroupName": "default",
      "DBSubnetGroupDescription": "default",
      "VpcId": "vpc-abcd0123",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-abcd0123",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-wxyz0123",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1b"
          },
          "SubnetStatus": "Active"
        }
      ]
    }
  ]
}
```

```
    },
    "PreferredMaintenanceWindow": "sun:06:53-sun:07:23",
    "PendingModifiedValues": {},
    "EngineVersion": "3.6.0",
    "AutoMinorVersionUpgrade": true,
    "PubliclyAccessible": false,
    "DBClusterIdentifier": "sample-cluster",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
    "DbiResourceId": "db-ABCDEFGHIJKLMNQPQRSTUVWXYZ",
    "CACertificateIdentifier": "rds-ca-2019",
    "PromotionTier": 1,
    "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance",
    "EnabledCloudwatchLogsExports": [
        "profiler"
    ]
}
}
```

執行個體重新啟動需要幾分鐘的時間來完成。您只能使用狀態為可用的執行個體。您可以使用主控台或 AWS CLI 監控執行個體的状态。如需詳細資訊，請參閱 [監控 Amazon DocumentDB 執行個體的状态](#)。

刪除 Amazon DocumentDB 實例

您可以使用 AWS Management Console 或刪除 Amazon DocumentDB 執行個體。AWS CLI 若要刪除執行個體，執行個體必須處於可用狀態。您無法刪除已停止的執行個體。如果包含執行個體的 Amazon DocumentDB 叢集停止，請先啟動叢集，等待執行個體可用，然後刪除執行個體。如需詳細資訊，請參閱 [停止和啟動 Amazon DocumentDB 集群](#)。

Note

Amazon DocumentDB 會將您的所有資料存放在叢集磁碟區中。資料會保留在該叢集磁碟區中，即使您移除叢集中的所有執行個體也一樣。如果您需要再次存取資料，可以隨時新增執行個體至叢集，並從您上次離開的位置繼續操作。

Using the AWS Management Console

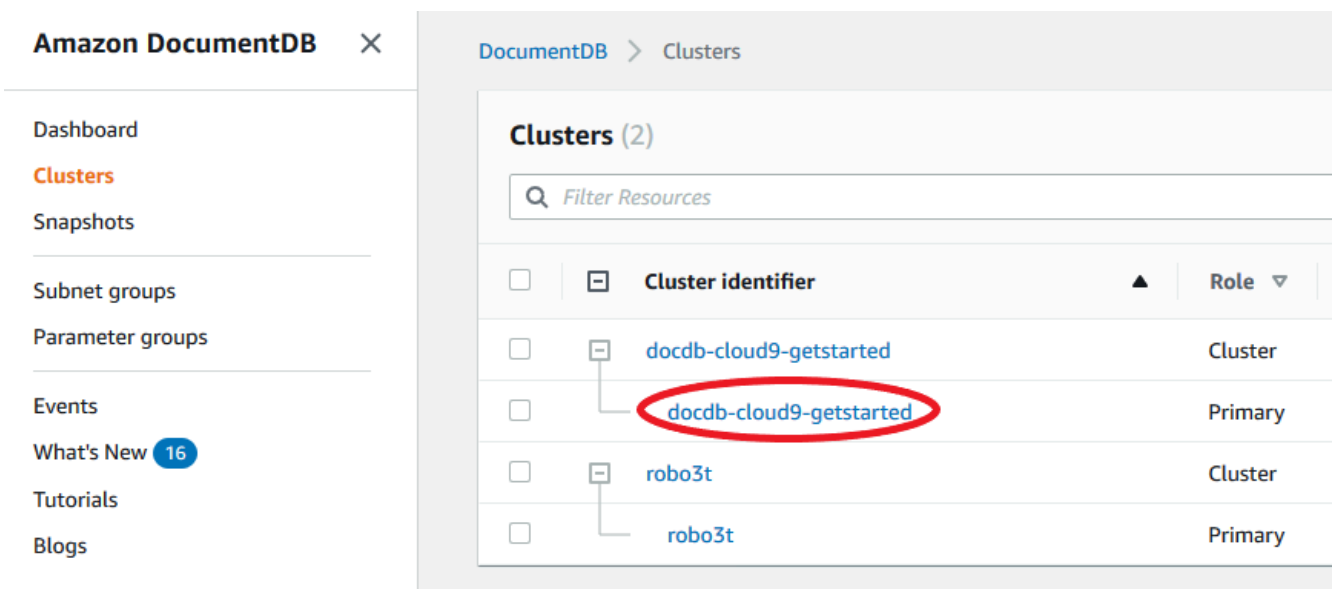
下列程序會使用主控台刪除指定的 Amazon DocumentDB 執行個體。

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 [叢集] 導覽方塊中，您會看到 [叢集識別碼] 資料行。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。



4. 核取您要刪除之執行個體左側的核取方塊。
5. 選取 Actions (動作)，然後選擇 Delete (刪除)。
 1. 如果您要刪除叢集中的最後一個執行個體：
 - Create final cluster snapshot? (建立最終叢集快照?) — 如果您要在刪除叢集之前建立最終快照，請選擇 [是]。否則請選擇 No (否)。
 - 最終快照名稱 — 如果您選擇建立最終快照，請輸入所建立之新叢集快照的叢集快照識別碼。
 - Delete <instance-name> instance? (刪除 <instance-name> 執行個體?) — 在欄位中輸入刪除整個叢集的短語以確認刪除。
 2. 如果您不是刪除叢集中的最後一個執行個體：

- Delete <instance-name> instance? (刪除 <instance-name> 執行個體?) — 在字段中輸入短語刪除我以確認刪除。

6. 選取 Delete (刪除) 以刪除執行個體。

刪除執行個體需要幾分鐘的時間。若要監控執行個體的状态，請參閱[監控 Amazon DocumentDB 執行個体的状态](#)。

Using the AWS CLI

下列程序會使用刪除 Amazon DocumentDB 執行個體。AWS CLI

1. 首先，請確定 Amazon DocumentDB 叢集中有多少個執行個體：

若要判斷叢集中有多少執行個體，請執行 describe-db-clusters 命令，如下所示。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].  
[DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
```

此操作的輸出將會如下所示。

```
[  
  [  
    "sample-cluster",  
    [  
      "sample-instance-1",  
      "sample-instance-2"  
    ]  
  ]  
]
```

2. 如果您的 Amazon DocumentDB 叢集中有多個執行個體：

若要刪除指定的 Amazon DocumentDB 執行個體，請使用具有 --db-instance-identifier 參數的 delete-db-instance 命令，如下所示。刪除執行個體需要幾分鐘的時間。若要監控執行個體的状态，請參閱[監控 Amazon DocumentDB 執行個体的状态](#)。

```
aws docdb delete-db-instance \  
  --db-instance-identifier sample-instance-2
```


此操作的輸出將會如下所示。

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "sample-instance-2",
    "DBInstanceClass": "db.r5.large",
    "Engine": "docdb",
    "DBInstanceStatus": "deleting",
    "Endpoint": {
      "Address": "sample-instance-2.node.us-east-1.docdb.amazonaws.com",
      "Port": 27017,
      "HostedZoneId": "ABCDEFGHIJKLM"
    },
    "InstanceCreateTime": "2020-03-27T08:05:56.314Z",
    "PreferredBackupWindow": "02:00-02:30",
    "BackupRetentionPeriod": 1,
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-abcd0123",
        "Status": "active"
      }
    ],
    "AvailabilityZone": "us-east-1c",
    "DBSubnetGroup": {
      "DBSubnetGroupName": "default",
      "DBSubnetGroupDescription": "default",
      "VpcId": "vpc-6242c31a",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-abcd0123",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-wxyz0123",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1b"
          },
          "SubnetStatus": "Active"
        }
      ]
    }
  }
}
```

```
    ]
  },
  "PreferredMaintenanceWindow": "sun:06:53-sun:07:23",
  "PendingModifiedValues": {},
  "EngineVersion": "3.6.0",
  "AutoMinorVersionUpgrade": true,
  "PubliclyAccessible": false,
  "DBClusterIdentifier": "sample-cluster",
  "StorageEncrypted": true,
  "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
  "DbiResourceId": "db-ABCDEFGHIJKLMNQPQRSTUVWXYZ",
  "CACertificateIdentifier": "rds-ca-2019",
  "PromotionTier": 1,
  "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2",
  "EnabledCloudwatchLogsExports": [
    "profiler"
  ]
}
}
```

3. 如果您要刪除的執行個體是 Amazon DocumentDB 叢集中的最後一個執行個體：

如果刪除 Amazon DocumentDB 叢集中的最後一個執行個體，也會刪除該叢集以及與該叢集相關聯的自動快照和連續備份。

若要刪除叢集中的最後一個執行個體，您可以刪除該叢集並選擇性地建立最終快照。如需詳細資訊，請參閱 [刪除 Amazon DocumentDB 集群](#)。

刪除保護

刪除 Amazon DocumentDB 叢集的最後一個執行個體也會刪除叢集，以及與該叢集相關聯的自動快照和連續備份。無論您是使用或執行刪除操作，Amazon DocumentDB 都會為叢集強制執行刪除保護。AWS Management Console AWS CLI 如果已啟用刪除保護，您就無法刪除叢集。

若要刪除已啟用刪除保護的叢集，您必須先修改叢集並停用刪除保護。如需更多詳細資訊，請參閱 [刪除 Amazon DocumentDB 集群](#)。

管理 Amazon DocumentDB 子網路群組

Virtual Private Cloud (VPC) 是您的 AWS 帳戶。其在邏輯上與 AWS 雲端的其他虛擬網路隔離。您可以在您的 Amazon VPC 中啟 Amazon DocuAWS 資料庫叢集。您可以為 VPC 指定 IP 地址範圍、新增子網路、與安全群組建立關聯，以及設定路由表。

子網是您 Amazon VPC 中的 IP 地址範圍。您可以在指定的子網中啟動 AWS 資源。針對必須連接到網際網路的資源使用「公有」子網路。針對不會連線至網際網路的資源使用私有子網。如需公有和私有子網路的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 和子網路基本](#)。

資料庫子網路群組是您在 VPC 中建立，然後指定給叢集之子網路的集合。子網路群組可讓您在建立叢集時指定特定的 VPC。如果您使用 default 子網路群組，它將橫跨 VPC 中的所有子網路。

各個資料庫子網路群組在指定區域中，都應在至少兩個可用區域中設有子網路。在 VPC 中建立資料庫叢集時，您必須選擇資料庫子網路群組。Amazon DocumentDB 要與叢集建立關聯的子網路群組。如果主要執行個體。然後，它可以使用前一個主要執行個體所在之子網路的 IP 地址，建立新的複本執行個體。

Amazon DocumentDB 在 VPC 中建立執行個體時，會使用從您資料庫子網路群組的 IP 地址，將網路介面指派給您的叢。我們強烈建議您使用 DNS 名稱，因為基本的 IP 地址可能會在容錯移轉期間有所變動。如需詳細資訊，請參閱 [Amazon DocumentDB 端點](#)。

如需有關建立自己的 VPC 和子網路的資訊，請參閱 [Amazon 虛擬私有雲使用者指南中的使用 VPC 和子網路](#)。

主題

- [建立 Amazon DocumentDB 子網路群組](#)
- [描述 Amazon DocumentDB 子網路群組](#)
- [修改 Amazon DocumentDB 子網路群組](#)
- [刪除 Amazon DocumentDB 子網路群組](#)

建立 Amazon DocumentDB 子網路群組

建立 Amazon DocumentDB 叢集時，您必須在該 Amazon VPC 內選擇一個 Amazon VPC 和對應的子網路群組來啟動叢集。子網路會決定您要用來啟動執行個體的可用區域內的可用區域和 IP 範圍。

子網路群組是一組具名的子網路 (或 AZ)，可讓您指定要用於啟動 Amazon DocumentDB 執行個體的可用區域。例如，在具有三個執行個體的叢集中，建議每個執行個體都在不同的 AZ 中佈建，這樣會針對高可用性進行最佳化。因此，如果單一 AZ 發生故障，只會影響單一執行個體。

目前，Amazon DocumentDB 執行個體最多可佈建在三個 AZ 中。即使子網路群組有三個以上的子網路，您也只能使用其中三個子網路來建立 Amazon DocumentDB 叢集。因此，建議您在建立子網路群組時，只選擇要部署執行個體的三個子網路。

例如：創建一個群集，Amazon DocumentDB 選擇 AZ {1A, 1B 和 1C}。如果您嘗試在 AZ 建立執行個體 {1D} API 呼叫將會失敗。不過，如果您選擇建立執行個體，而不指定特定的可用區域，Amazon DocumentDB 會代表您選擇可用區域。Amazon DocumentDB 使用演算法來平衡 AZ 執行個體的執行個體，以協助您達到高可用性。如果佈建了三個執行個體，預設會在三個 AZ 之間佈建這些執行個體，而且不會在單一可用區中全部佈建。

最佳實務

- 除非有特定的原因，否則一律使用三個子網路建立子網路群組。這可確保具有三個或更多執行個體的叢集能夠達到更高的可用性，因為執行個體將佈建在三個 AZ 之間。
- 一律將執行個體分散在多個可用區域以達到高可用性。請勿將叢集的所有執行個體放置在單一可用區。
- 由於隨時都會發生容錯移轉事件，您不應假設主執行個體或複本執行個體會特定的可用區域。

如何建立子網路群組

您可以使用AWS Management Console或AWS CLI來建立 Amazon DocumentDB 子網路群組：

Using the AWS Management Console

使用下列步驟來建立 Amazon DocumentDB 網路群組。

建立 Amazon DocumentDB 子網路群組

1. 登入AWS Management Console，並前往 <https://console.aws.amazon.com/docdb> 開 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇 Subnet groups (子網路群組)，然後選擇 Create (建立)。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示

(☰

)。

3. 在 Create subnet group (建立子網路群組) 頁面上：

- a. 在 Subnet group details (子網路群組詳細資訊) 區段中：
 - i. 名稱。
 - ii. Description (描述)：輸入子網路群組的描述。
 - b. 在 Add subnets (新增子網路) 區段中：
 - i. VPC — 在清單中，選擇此子網路群組的 VPC。
 - ii. 執行下列任意一項：
 - 若要將所有子網路包含在所選的 VPC 中，請選擇 Add all the subnets related to this VPC (新增與此 VPC 相關的所有子網路)。
 - 若要指定此子網路群組的子網路，請在想要包含子網路的每個可用區域中執行下列動作。您必須至少包含兩個可用區域。
 - A. 可用區域 — 在清單中，選擇可用區域。
 - B. 子網路 — 在清單中，從此子網路群組選擇所選可用區域的子網路。
 - C. 選擇 Add subnet (新增子網路)。
4. 選擇 建立。建立子網路群組時，其會與您其他的子網路群組一起列出。

| Name | Description | Status | VPC |
|---------------------|-----------------------|----------|--------------|
| default | default | Complete | vpc-91280df6 |
| sample-subnet-group | A sample subnet group | Complete | vpc-91280df6 |

Using the AWS CLI

在能夠使用 AWS CLI 建立子網路群組之前，您必須先判斷哪些子網路可用。執行下列 AWS CLI 操作，列出可用區域及其子網路。

參數：

- **--db-subnet-group** 選擇性。指定特定的子網路群組，會列出該群組的可用區域和子網路。略過此參數，會列出您所有子網路群組的可用區域和子網路。指定 default 子網路群組，會列出所有 VPC 的子網路。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-subnet-groups \  
  --db-subnet-group-name default \  
  --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].  
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

針對 Windows：

```
aws docdb describe-db-subnet-groups ^  
  --db-subnet-group-name default ^  
  --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].  
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  [  
    "default",  
    [  
      [  
        "us-east-1a",  
        "subnet-4e26d263"  
      ],  
      [  
        "us-east-1c",  
        "subnet-afc329f4"  
      ],  
      [  
        "us-east-1e",  
        "subnet-b3806e8f"  
      ],  
      [  
        "us-east-1d",  
        "subnet-53ab3636"  
      ],  
      [  
        "us-east-1b",  
        "subnet-991cb8d0"  
      ],  
      [  

```

```

        "us-east-1f",
        "subnet-29ab1025"
    ]
]
]
]
]

```

使用先前操作的輸出可建立新的子網路群組。新的子網路群組必須包含至少兩個可用區域的子網路。

參數：

- **--db-subnet-group-name** - 必要項目。此子網路群組的名稱。
- **--db-subnet-group-description** - 必要項目。此子網路群組的描述。
- **--subnet-ids** - 必要項目。要包含在此子網路群組中之所有子網路的清單。範例：subnet-53ab3636。
- **-標籤**-可選。要連接到此子網路群組的標籤 (金鑰值對) 清單。

以下程式碼會建立子網路群組 `sample-subnet-group`，其含有三個子網路 `subnet-4e26d263`、`subnet-afc329f4` 和 `subnet-b3806e8f`。

若為 Linux、macOS 或 Unix：

```

aws docdb create-db-subnet-group \
  --db-subnet-group-name sample-subnet-group \
  --db-subnet-group-description "A sample subnet group" \
  --subnet-ids subnet-4e26d263 subnet-afc329f4 subnet-b3806e8f \
  --tags Key=tag1,Value=One Key=tag2,Value=2

```

針對 Windows：

```

aws docdb create-db-subnet-group ^
  --db-subnet-group-name sample-subnet-group ^
  --db-subnet-group-description "A sample subnet group" ^
  --subnet-ids subnet-4e26d263 subnet-afc329f4 subnet-b3806e8f ^
  --tags Key=tag1,Value=One Key=tag2,Value=2

```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
```

```
"DBSubnetGroup": {
  "DBSubnetGroupDescription": "A sample subnet group",
  "DBSubnetGroupName": "sample-subnet-group",
  "Subnets": [
    {
      "SubnetAvailabilityZone": {
        "Name": "us-east-1a"
      },
      "SubnetIdentifier": "subnet-4e26d263",
      "SubnetStatus": "Active"
    },
    {
      "SubnetAvailabilityZone": {
        "Name": "us-east-1c"
      },
      "SubnetIdentifier": "subnet-afc329f4",
      "SubnetStatus": "Active"
    },
    {
      "SubnetAvailabilityZone": {
        "Name": "us-east-1e"
      },
      "SubnetIdentifier": "subnet-b3806e8f",
      "SubnetStatus": "Active"
    }
  ],
  "VpcId": "vpc-91280df6",
  "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
  "SubnetGroupStatus": "Complete"
}
```

描述 Amazon DocumentDB 子網路群組

您可以使用AWS Management Console或取AWS CLI得 Amazon DocumentDB 子網路群組的詳細資料。

Using the AWS Management Console

下列程序示範如何取得 Amazon Docu資料庫子網路群組。

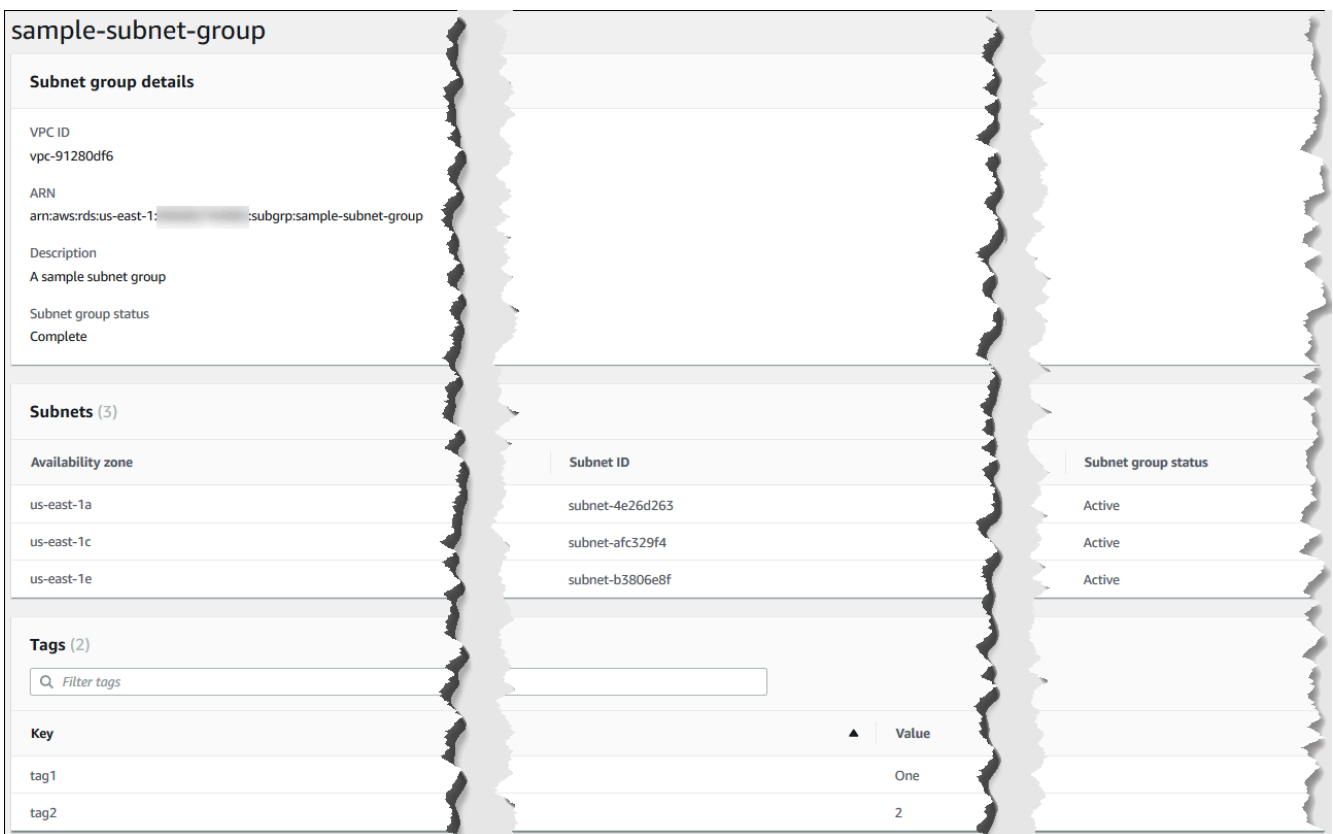
找出子網路群組的詳細資訊

1. 登入AWS Management Console，並前往 <https://console.aws.amazon.com/docdb> 開 Amazon DocumentDB 主控台。
2. 在導覽窗格中選擇 Subnet groups (子網路群組)。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 若要查看子網路群組的詳細資訊，請選擇該子網路群組的名稱。



The screenshot displays the AWS Management Console interface for a subnets group. The main content area is titled 'sample-subnet-group' and is divided into several sections:

- Subnet group details:** This section provides key information about the group:
 - VPC ID: vpc-91280df6
 - ARN: arn:aws:rds:us-east-1: [redacted]:subgrp:sample-subnet-group
 - Description: A sample subnet group
 - Subnet group status: Complete
- Subnets (3):** A table listing the subnets within the group:

| Availability zone | Subnet ID | Subnet group status |
|-------------------|-----------------|---------------------|
| us-east-1a | subnet-4e26d263 | Active |
| us-east-1c | subnet-afc329f4 | Active |
| us-east-1e | subnet-b3806e8f | Active |
- Tags (2):** A section for managing tags, including a search filter and a table:

| Key | Value |
|------|-------|
| tag1 | One |
| tag2 | 2 |

Using the AWS CLI

若要尋找 Amazon DocumentDB 子網路群組的詳細資訊，請使用具有下列參數的 `describe-db-subnet-groups` 作業。

參數

- `--db-subnet=group-name` 選擇性。如果包含，將列出指名子網路群組的詳細資訊。如果省略，將列出最多 100 個子網路群組的詳細資訊。

Example

以下程式碼將列出我們在第 [建立 Amazon DocumentDB 子網路群組](#) 節中建立的 `sample-subnet-group` 子網路群組的詳細資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-subnet-groups \  
  --db-subnet-group-name sample-subnet-group
```

針對 Windows：

```
aws docdb describe-db-subnet-groups ^  
  --db-subnet-group-name sample-subnet-group
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBSubnetGroup": {  
    "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-subnet-group",  
    "VpcId": "vpc-91280df6",  
    "SubnetGroupStatus": "Complete",  
    "DBSubnetGroupName": "sample-subnet-group",  
    "Subnets": [  
      {  
        "SubnetAvailabilityZone": {  
          "Name": "us-east-1a"  
        },  
        "SubnetStatus": "Active",  
        "SubnetIdentifier": "subnet-4e26d263"  
      },  
      {  
        "SubnetAvailabilityZone": {  
          "Name": "us-east-1c"  
        },  
      }  
    ]  
  }  
}
```

```
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-afc329f4"
    },
    {
        "SubnetAvailabilityZone": {
            "Name": "us-east-1e"
        },
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-b3806e8f"
    }
],
"DBSubnetGroupDescription": "A sample subnet group"
}
```

修改 Amazon DocumentDB 子網路群組

您可以使用AWS Management Console或修AWS CLI改子網路群組的說明，或從 Amazon DocumentDB 子網路群組新增或移除子網路。不過，您無法修改 default 子網路群組。

Using the AWS Management Console

您可以使用 AWS Management Console 變更子網路群組的描述，或新增和移除子網路。請記住，完成後，您必須至少有兩個可用區域與子網路群組關聯。

修改子網路群組

1. 登入AWS Management Console，並前往 <https://console.aws.amazon.com/docdb> 開 Amazon DocumentDB 主控台。
2. 在導覽窗格中選擇 Subnet groups (子網路群組)。然後，選擇子網路群組名稱左側的按鈕。請記住，您無法修改 default 子網路群組。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 選擇 Actions (動作)，然後選擇 Modify (修改)。
4. 說明 — 若要變更子網路群組的描述，請輸入新的說明。

5. 若要變更與子網路群組相關聯的子網路，請在 Add subnets (新增子網路) 區段中執行下列其中一個或多個動作：
 - 若要移除此子網路群組中的所有子網路，選擇 Remove all (全部移除)。
 - 若要從此子網路群組中移除特定的子網路，請為每個想要移除的子網路選擇 Remove (移除)。
 - 若要新增與此 VPC 相關的所有子網路，請選擇 Add all the subnets related to this VPC (新增與此 VPC 相關的所有子網路)。
 - 若要新增特定子網路至子網路群組，請為想要新增子網路的每個可用區域執行下列動作。
 - a. 可用區域 — 在清單中，選擇新的可用區域。
 - b. 子網路 — 在清單中，從此子網路群組選擇所選可用區域的子網路。
 - c. 選擇 Add subnet (新增子網路)。
6. 在確認對話方塊中：
 - 若要對子網路群組進行這些變更，請選擇 Modify (修改)。
 - 若要保持子網路群組不變，請選擇 Cancel (取消)。

Using the AWS CLI

您可以使用 AWS CLI 變更子網路群組的描述，或新增和移除子網路。請記住，完成後，您必須至少有兩個可用區域與子網路群組關聯。您無法修改 default 子網路群組。

參數：

- `--db-subnet-group-name` - 必要項目。您正在修改的 Amazon DocumentDB 子網路群組名稱。
- `--subnet-ids` - 必要項目。完成此變更之後，您想要包含在子網路群組內的所有子網路的清單。

Important

目前在子網路群組中但未包含在此清單內任何子網路將從子網路群組移除。如果您想要保留目前在子網路群組中的任何子網路，您必須將其加入到清單內。

- `--db-subnet-group-description` 選擇性。子網路群組的描述。

Example

以下程式碼將修改描述並用子網路 subnet-991cb8d0、subnet-53ab3636 及 subnet-29ab1025 取代現有的子網路。

若為 Linux、macOS 或 Unix：

```
aws docdb modify-db-subnet-group \  
  --db-subnet-group-name sample-subnet-group \  
  --subnet-ids subnet-991cb8d0 subnet-53ab3636 subnet-29ab1025 \  
  --db-subnet-group-description "Modified subnet group"
```

針對 Windows：

```
aws docdb modify-db-subnet-group ^  
  --db-subnet-group-name sample-subnet-group ^  
  --subnet-ids subnet-991cb8d0 subnet-53ab3636 subnet-29ab1025 ^  
  --db-subnet-group-description "Modified subnet group"
```

此操作的輸出將會如下所示 (JSON 格式)。請注意，這是已在[建立 Amazon DocumentDB 子網路群組](#)一節中建立的同一個子網路群組。不過，子網路群組的子網路會取代為 modify-db-subnet-group 操作中列出的子網路。

```
{  
  "DBSubnetGroup": {  
    "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-subnet-group",  
    "DBSubnetGroupDescription": "Modified subnet group",  
    "SubnetGroupStatus": "Complete",  
    "Subnets": [  
      {  
        "SubnetAvailabilityZone": {  
          "Name": "us-east-1d"  
        },  
        "SubnetStatus": "Active",  
        "SubnetIdentifier": "subnet-53ab3636"  
      },  
      {  
        "SubnetAvailabilityZone": {  
          "Name": "us-east-1b"  
        },  
      }  
    ]  
  }  
}
```

```
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-991cb8d0"
    },
    {
        "SubnetAvailabilityZone": {
            "Name": "us-east-1f"
        },
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-29ab1025"
    }
],
"VpcId": "vpc-91280df6",
"DBSubnetGroupName": "sample-subnet-group"
}
```

刪除 Amazon DocumentDB 子網路群組

您可以使用AWS Management Console或AWS CLI來刪除 Amazon DocumentDB 網路群組。不過，您無法刪除 default 子網路群組。

Using the AWS Management Console

您可以使用 AWS Management Console 刪除子網路群組。但是，您無法刪除 default 子網路群組。

刪除子網路群組

1. 登入AWS Management Console，並前往 <https://console.aws.amazon.com/docdb> 開 Amazon DocumentDB 主控台。
2. 在導覽窗格中選擇 Subnet groups (子網路群組)。然後，選擇子網路群組名稱左側的按鈕。請注意，您無法刪除 default 子網路群組。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 選擇 動作，然後選擇 刪除。
4. 在確認對話方塊中：

- 若要刪除子網路群組，請選擇 Delete (刪除)。
- 若要保留子網路群組，請選擇 Cancel (取消)。

Using the AWS CLI

若要使用刪除 Amazon DocumentDB 子網路群組 AWS CLI，請使用具有下列參數的 `delete-db-subnet-group` 作業。

參數

- `--db-subnet-group-name` - 必要項目。要刪除的 Amazon DocumentDB 子網路群組。請注意，您無法刪除 `default` 子網路群組。

Example

以下程式碼會刪除 `sample-subnet-group`。

若為 Linux、macOS 或 Unix：

```
aws docdb delete-db-subnet-group \  
  --db-subnet-group-name sample-subnet-group
```

針對 Windows：

```
aws docdb delete-db-subnet-group ^  
  --db-subnet-group-name sample-subnet-group
```

此操作不會產生輸出。

Amazon DocumentDB 高可用性和複寫

您可以透過使用複寫執行個體來達成在 Amazon DocumentDB 中的高可用性和讀取擴展（具有 MongoDB 兼容性）。單一 Amazon DocumentDB 叢集支援單一主要執行個體，以及最多 15 個複本執行個體。這些執行個體可以分佈在叢集區域內的可用區域之中。主要執行個體接受讀取和寫入流量，而複寫執行個體僅接受讀取請求。

叢集磁碟區是由叢集的多個資料複本組成。不過，叢集磁帶區中的資料是以單一邏輯磁盤區的形式呈現給主要執行個體，以及叢集中的 Amazon DocumentDB 複本。複本執行個體是最終一致性 它們會透過

最低複寫延遲 (通常在主要執行個體寫入更新後短於 100 毫秒) 來傳回查詢結果。複本延遲會根據資料庫變更率而有所不同。也就是，在資料庫發生大量寫入操作期間，您可能會發現複本延遲增加。

讀取擴展

Amazon DocumentDB 複本很適用於讀取擴展，因為完全專用於叢集磁盤區上的讀取操作。寫入操作由主要執行個體管理。叢集中的所有執行個體將共用叢集磁碟區。因此，您不必複寫和維護每個 Amazon DocumentDB 複本的資料副本。

高可用性

當您建立 Amazon DocumentDB 叢集時，根據子網路組中的可用區域數量 (必須至少兩個)，Amazon DocumentDB 會跨可用區域佈建執行個體。當您在叢集中建立執行個體時，Amazon DocumentDB 會自動將執行個體分佈在子網路叢集的可用區域之中，以平衡叢集。這個動作還可防止所有執行個體位於相同的可用區域。

範例

為說明要點，假設我們建立含有一個子網路組，且該子網路組有三個可用區域的叢集：AZ1、AZ2，以及AZ3。

當叢集的第一個執行個體建立完成後，其為主執行個體，且位於其中一個可用區域。在此範例中，其位於 AZ1。建立的第二個執行個體為複本執行個體，其位於另外兩個可用區域其中之一，假設為 AZ2。建立的第三個執行個體為複本執行個體，其位於剩餘的可用區域，即 AZ3。如果您建立多個執行個體，它們會分佈在可用區域之中，讓您在叢集中達到平衡。

如果故障發生在主執行個體 (AZ1)，將觸發容錯移轉，且其中一個現有的複本會提升至主要執行個體。當舊的主要執行個體恢復時，它會在佈建 (AZ1) 的相同可用區域中成為複本。當佈建有三個執行個體的叢集時，Amazon DocumentDB 會繼續保留有三個執行個體的叢集。Amazon DocumentDB 會自動處理偵測、容錯移轉，和恢復執行個體故障，無需任何手動幹預。

當 Amazon DocumentDB 執行容錯移轉並恢復執行個體時，恢復的執行個體會保留在原先佈建的可用區域。不過，執行個體角色可能從主要變更為複本。這樣做可防止一連串的容錯移轉導致所有執行個體出現在同一個可用區域內。

您可以指定 Amazon DocumentDB 複本為容錯移轉目標。也就是說，如果主要執行個體失敗，指定的 Amazon DocumentDB 複本或來自某層的複本將提升為主要執行個體。在對主要執行個體提出的讀取和寫入請求由於例外狀況而失敗的期間，會發生短暫的中斷。如果 Amazon DocumentDB 叢集不包含任何 Amazon DocumentDB 複本，則當主要執行個體失敗時，則會重新建立主要執行個體。提升 Amazon DocumentDB 複本遠比重重新建立主要執行個體快得多。

對於高可性案例，我們建議您建立一個或多個 Amazon DocumentDB 複本。這些複本的執行個體類別應與主要執行個體相同，且位於與 Amazon DocumentDB 叢集不同的可用區域。

如需詳細資訊，請參閱下列內容：

- [了解 Amazon DocumentDB 叢集容錯](#)
- [Amazon DocumentDB 容錯移轉](#)
 - [控制容錯移轉目標](#)

具有全局羣集的高可用性

實現跨多個AWS 區域，您可以設定[Amazon DocumentDB 全球叢集](#)。每個全域叢集橫跨多個區域，可實現各個AWS 區域。Amazon DocumentDB 會自動處理從主要區域到每個次要區域的所有資料和更新的複寫。

新增 複本

叢集中新增的第一個執行個體即為主要執行個體。在第一個執行個體後新增的每個執行個體即為複寫執行個體。叢集最多可以有 15 個複本執行個體，以及主要執行個體。

當您使用 AWS Management Console 建立叢集時，會同時自動建立主要執行個體。若要在您建立叢集和主要執行個體的同時建立複本，請選擇 Create replica in different zone (在不同區域中建立複本)。如需詳細資訊，請參閱 [創建一個 Amazon DocumentDB 集群](#) 中的步驟 4.d。若要將更多複本新增至 Amazon DocumentDB 叢集，請參閱 [將 Amazon DocumentDB 執行個體新增至叢集](#)。

使用 AWS CLI 來建立叢集時，您必須明確建立主要和複寫執行個體。如需詳細資訊，請參閱以下主題中的「使用 AWS CLI」一節：

- [創建一個 Amazon DocumentDB 集群](#)
- [將 Amazon DocumentDB 執行個體新增至叢集](#)

Amazon DocumentDB 容錯移轉

在某些情況下，例如特定類型的計劃維護，或者在主要節點或可用區域故障時，Amazon DocumentDB (具有 MongoDB 兼容性) 會檢測容錯移轉並替換主要節點。容錯移轉期間，寫入時間會降至最低。這是因為主要節點的角色會容錯移轉至其中一個僅供讀取複本，而非必須建立並佈建新的主要節點。此故障偵測及複本提升可確保您能在提升完成時立即繼續寫入新的主要節點。

若要讓容錯移轉正常運作，您的叢集至少必須有兩個執行個體 — 一個主要執行個體和至少一個複本執行個體。

控制容錯移轉目標

Amazon DocumentDB 為您提供容錯移轉方案，作為容錯移轉發生時，控制將哪個複本執行個體提升成主要執行個體的方法。

容錯移轉方案

每個複本執行個體都會關聯至一個容錯移轉方案 (0—15)。如果因為維護或不太可能出現的硬件故障而發生容錯移轉，主要執行個體會容錯移轉至優先順序最高的複本 (編號最小的方案)。如果多個複本有相同的優先順序方案，主要執行個體會容錯移轉至該層的複本，該複本大小最接近上一個主複本。

選擇一組複本，並將其容錯移轉方案設定為 0 (最高優先順序)，就可以確保容錯移轉會提升到該群組的其中一個複本。為複本指派低優先順序方案 (數字較大)，就可以有效防止容錯移轉發生時，將特定複本提升成主要執行個體。這在特定複本接收來自應用程式的繁重使用情形時非常有用，容錯移轉至這些複本的其中一個將會對關鍵應用程式造成負面影響。

您可以在建立執行個體時設定容錯移轉方案，或在稍後進行修改。藉由修改執行個體來設定執行個體容錯移轉方案，並不會觸發容錯移轉。如需詳細資訊，請參閱下列主題：

- [將 Amazon DocumentDB 執行個體新增至叢集](#)
- [修改 Amazon DocumentDB 實例](#)

手動啟動容錯移轉時，您有兩種方式來控制要將哪個執行個體複本提升為主要執行個體：透過前面所述的容錯移轉方案和 `--target-db-instance-identifier` 參數。

`--target-db-instance-identifier`

若要進行測試，您可以使用 `failover-db-cluster` 操作強制容錯移轉事件發生。您可以使用 `--target-db-instance-identifier` 參數來指定將哪些複本提升成主要執行個體。使用 `--target-db-instance-identifier` 參數會取代容錯移轉優先順序方案。如果您未指定 `--target-db-instance-identifier` 參數，主要容錯移轉會根據容錯移轉優先順序方案。

容錯移轉期間會發生什麼情況？

Amazon DocumentDB 會自動處理容錯移轉，讓您的應用程式可以盡快恢復資料庫操作，而無須人為介入管理。

- 如果在故障轉移時，如果您在相同或不同的可用區域中有 Amazon DocumentDB 副本實例：Amazon DocumentDB 會翻轉執行個體的正式名稱記錄 (CNAME) 以指向執行狀態正常的複本，該複本轉而提升成新的主複本。容錯移轉從開始到結束通常可在 30 秒內完成。
- 如果您沒有 Amazon DocumentDB 複本執行個體 (例如，單個執行個體叢集)，請執行以下操作：Amazon DocumentDB 會嘗試在與原始執行個體所在的相同可用區域內建立新的執行個體。已盡力進行這種原始執行個體的取代操作，但可能不成功，例如，在出現會廣泛影響可用區域的問題時。

您的應用程式應該會在發生連線中斷時重試資料庫連線。

測試容錯移轉

叢集的容錯移轉會將叢集中的其中一個 Amazon DocumentDB 複本 (只讀執行個體) 提升成主要執行個體 (叢集寫入器)。

主要執行個體故障時，Amazon DocumentDB 會自動容錯移轉至 Amazon DocumentDB 複本 (如果有的話)。當您想要模擬主要執行個體故障以進行測試時，可以強制容錯移轉。叢集中的每個執行個體都有自己的端點地址。因此，當容錯移轉完成時，您需要清除和重新建立任何使用這些端點地址的現有連線。

若要強制容錯移轉，請搭配這些參數使用 `failover-db-cluster` 操作。

- `--db-cluster-identifier` - 必要項目。要容錯移轉之叢集的名稱。
- `--target-db-instance-identifier`— 選用。要提升為主要執行個體之執行個體的名稱。

Example

以下操作會強制容錯移轉 `sample-cluster` 叢集。Amazon DocumentDB 會根據容錯移轉方案優先順序選擇執行個體。

針對 Linux、macOS 或 Unix：

```
aws docdb failover-db-cluster \  
  --db-cluster-identifier sample-cluster
```

針對 Windows：

```
aws docdb failover-db-cluster ^  
  --db-cluster-identifier sample-cluster
```

以下操作會強制容錯移轉 `sample-cluster` 叢集，指定 `sample-cluster-instance` 為要提升的主要角色。(請注意輸出中的 `"IsClusterWriter": true`。)

針對 Linux、macOS 或 Unix：

```
aws docdb failover-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --target-db-instance-identifier sample-cluster-instance
```

針對 Windows：

```
aws docdb failover-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --target-db-instance-identifier sample-cluster-instance
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBCluster": {  
    "HostedZoneId": "Z2SUY0A1719RZT",  
    "Port": 27017,  
    "EngineVersion": "3.6.0",  
    "PreferredMaintenanceWindow": "thu:04:05-thu:04:35",  
    "BackupRetentionPeriod": 1,  
    "ClusterCreateTime": "2018-06-28T18:53:29.455Z",  
    "AssociatedRoles": [],  
    "DBSubnetGroup": "default",  
    "MasterUsername": "master-user",  
    "Engine": "docdb",  
    "ReadReplicaIdentifiers": [],  
    "EarliestRestorableTime": "2018-08-21T00:04:10.546Z",  
    "DBClusterIdentifier": "sample-cluster",  
    "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",  
    "DBClusterMembers": [  
      {  
        "DBInstanceIdentifier": "sample-cluster-instance",  
        "DBClusterParameterGroupStatus": "in-sync",  
        "PromotionTier": 1,  
        "IsClusterWriter": true  
      },  
      {  
        "DBInstanceIdentifier": "sample-cluster-instance-00",  
        "DBClusterParameterGroupStatus": "in-sync",
```

```
        "PromotionTier": 1,
        "IsClusterWriter": false
    },
    {
        "DBInstanceIdentifier": "sample-cluster-instance-01",
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1,
        "IsClusterWriter": false
    }
],
"AvailabilityZones": [
    "us-east-1b",
    "us-east-1c",
    "us-east-1a"
],
"DBClusterParameterGroup": "default.docdb3.6",
"Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
"IAMDatabaseAuthenticationEnabled": false,
"AllocatedStorage": 1,
"LatestRestorableTime": "2018-08-22T21:57:33.904Z",
"PreferredBackupWindow": "00:00-00:30",
"StorageEncrypted": false,
"MultiAZ": true,
"Status": "available",
"DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
"VpcSecurityGroups": [
    {
        "Status": "active",
        "VpcSecurityGroupId": "sg-12345678"
    }
],
"DbClusterResourceId": "cluster-ABCDEFGHIJKLMNQRSTUWXYZ"
}
}
```

複寫延遲

複製延遲通常為 50 毫秒或更少。複本延遲增加的最常見原因為：

- 主服務器上的高寫入速率，導致只讀副本落在主數據庫後面。
- 長時間運行的查詢（例如，大型順序掃描、聚合查詢）和傳入寫複製之間的只讀副本爭用。
- 只讀副本上的大量併發查詢。

要最大限度地減少複製滯後，請嘗試以下故障排除技術

- 如果您的寫入速率較高或 CPU 利用率較高，我們建議您在集羣中擴展實例。
- 如果只讀副本上運行時間較長的查詢，並且要查詢的文檔非常頻繁地更新，請考慮更改長時間運行的查詢，或者針對主/寫副本運行查詢，以避免只讀副本爭用。
- 如果只讀副本上存在大量併發查詢或 CPU 使用率高，則另一種選擇是擴展只讀副本的數量以分散工作負載。
- 由於複製滯後是高寫入吞吐量和長時間運行的查詢導致的結果，因此我們建議您使用 DBClusterReplicaLagMagMaMagMaCW 度量與慢查詢記錄器和 WriteThroughput/WriteIOPS 指標。

通常，我們建議您的所有副本都具有相同的實例類型，以便羣集故障轉移不會導致性能下降。

如果您在向上擴展和橫向擴展之間進行選擇（例如，六個較小的實例與三個較大的實例），我們通常建議在擴展之前先嘗試向上擴展（較大的實例），因為每個數據庫實例將獲得更大的緩衝區緩存。

主動地，您應該設置複製滯後警報，並將其閾值設置為您認為在副本實例上的數據可能落後（或「過時」）的上限，然後再開始影響應用程序的功能。一般來說，由於工作負載暫時，我們建議在發出警報之前超過多個數據點的複製滯後閾值。

Note

此外，我們建議您為超過 10 秒的複製滯後設置另一個警報。如果多個數據點超過此閾值，我們建議您向上擴展實例或降低主實例上的寫入吞吐量。

管理 Amazon DocumentDB 索引

Amazon DocumentDB 索引創建

在 Amazon DocumentDB 中構建索引需要做出一些決策：

- 它需要多快完成？
- 在構建發生時，集合是否無法訪問？
- 可以將多少執行個體運算能力分配給組建？
- 應該創建什麼類型的索引？

本節可協助您回答這些問題，並提供在執行個體型叢集上建立 Amazon DocumentDB 索引的命令和監控範例。

指導方針

下列準則包括建立新索引時的基本限制和組態權衡：

- Amazon DocumentDB 版本支援-雖然所有 Amazon DocumentDB 版本都支援單一工作者索引，但只有 Amazon DocumentDB 版本 4.0 和 5.0 版本才支援多個工作者索引。
- 效能權衡-增加索引建立程序中的工作者數量可增加 CPU 使用率，並讀取 Amazon DocumentDB 資料庫主要執行個體上的 IO。建立新索引所需的資源不適用於執行中的工作負載。
- 彈性叢集-Amazon DocumentDB 彈性叢集不支援並行索引。
- Worker 數目上限-您可以設定的 Worker 數目上限取決於資料庫叢集中主要執行個體的大小。它是資料庫叢集主要執行個體上 vCPUs 總數的一半。例如，您可以在具有 64 個 vCPUs 的資料庫 .r6g.16xlarge 執行個體上執行最多 32 個工作程式。

Note

2xlarge 執行個體類別及更低版本不支援並行工作程式。

- 最少工作人數-您可以配置的最少工作者數量為 1。在執行個體型叢集上建立索引的預設設定為兩個 Worker。但是，您可以通過使用「工作線程」選項將 Worker 數量減少為一個。這將使用單個 Worker 運行該進程。
- 索引壓縮-Amazon DocumentDB 不支持索引壓縮。索引的資料大小可能會比您使用其他選項時大。
- 索引多個集合-資料庫叢集主要執行個體上一半的 vCPUs 可用於在多個集合上執行索引建立的已設定背景工作者。
- 索引類型-如需 Amazon DocumentDB 上支援索引類型的完整說明，請參閱[此部落格文章](#)。

開始使用

若要在集合上開始建立索引，請使用指 `createIndexes` 令。根據預設，命令會執行兩個 parallel Worker，將索引建立程序的速度提高兩倍。

例如，下列命令程序示範如何為文件中的「user_name」欄位建立索引，並將索引程序速度提高至四個 Worker：

1. 使用叢集上的兩個 parallel Worker 建立索引：

```
db.runCommand({"createIndexes":"test","indexes":[{"key": {"user_name":1},
"index_name":"username_idx"}]})
```

- 若要最佳化索引建立程序的速度，您可以使用指 `db.runCommand createIndexes` 令中的「Worker thread」選項 (`"workers":<number>`) 來指定 Worker 數目。

將流程的速度提高到四個 parallel 工作者：

```
db.runCommand({"createIndexes":"test","indexes":[{"key": {"user_name":1},
"index_name":"username_idx", "workers":4}]}))
```

Note

Worker 的數目越多，索引建立的進度就越快。不過，Worker 數目增加越多，vCPUs 和讀取主執行個體的 IO 負載就越高。確保您的叢集已充分佈建，以處理增加的負擔，而不會降低其他工作負載。

索引進度狀態

索引建立程序的運作方式是初始化、掃描集合、排序索引鍵，以及最後透過索引產生器插入索引鍵。當您在前台執行程序時，處理序最多有六個階段，以及最多 9 個階段，當您在背景執行。您可以逐階段檢視狀態指標，例如完成百分比、已掃描儲存區塊的總數、已排序的金鑰，以及插入的金鑰。

通過使用 mongo shell 中的 `db.currentOp()` 命令監視索引過程的進度。最後一個階段的 100% 完成表明所有索引都已成功創建：

```
db.currentOp({"command.createIndexes": { $exists : true } })
```

索引構建類型

索引建置的四種類型為：

- 前景-前景索引構建塊所有其他數據庫操作，直到創建索引。Amazon DocumentDB 前台構建由五個階段組成。
- 前景 (唯一) - 單個文檔 (唯一) 前景索引構建阻止其他數據庫操作，如常規前台構建。與基本的前景構建不同，唯一構建使用額外的階段 (排序鍵 2) 來查找重複的鍵。前景 (唯一) 組建由六個階段組成。

- 背景-背景索引構建允許在創建索引時在前台運行其他數據庫操作。Amazon DocumentDB 背景構建由八個階段組成。
- 背景 (唯一) -單個文檔 (唯一) 背景索引構建允許在創建索引時在前台運行其他數據庫操作。與基本的背景構建不同，獨特的構建使用額外的階段 (排序鍵 2) 來查找重複的鍵。背景 (唯一) 構建由九個階段組成。

索引構建階段

| 階段 | 前景 | 前景 (唯一) | 背景介紹 | 背景 (獨特) |
|------------|----|---------|------|-----------|
| 初始化 | 1 | 1 | 1 | 1 |
| 建築索引:初始化 | 2 | 2 | 2 | 2 |
| 建築索引:掃描收藏 | 3 | 3 | 3 | 3 |
| 建築索引：排序鍵 1 | 4 | 4 | 4 | 4 |
| 建築索引：排序鍵 2 | | 5 | | 5 |
| 建築索引：插入鍵 | 5 | 6 | 5 | 6 |
| 驗證:掃描索引 | | | 6 | 7 |
| 驗證:排序元組 | | | 7 | 8 |
| 驗證:掃描收集 | | | 8 | 9 |

- 初始化-創建索引正在準備索引生成器。這個階段應該非常簡短。
- 建設索引：初始化-索引生成器正準備創建索引。這個階段應該非常簡短。
- 建築索引：掃描收集-索引生成器正在執行集合掃描以收集索引鍵。測量單位是「塊」。

Note

如果為索引建置配置了多個 Worker，則會在此階段中顯示該 Worker。「掃描收集」階段是唯一在索引建置過程中使用多個 Worker 的階段。所有其他階段都會顯示單一 Worker。

- 建設索引：排序鍵 1-索引生成器正在對收集的索引鍵進行排序。度量單位是「鑰匙」。
- 構建索引：排序鍵 2-索引生成器正在對應於死元組的收集索引鍵進行排序。此階段僅適用於唯一索引建立。度量單位是「鑰匙」。
- 建設索引：插入鍵-索引生成器正在將索引鍵插入到新索引中。度量單位是「鑰匙」。
- 驗證：掃描索引-createIndex 正在掃描索引以查找需要驗證的密鑰。測量單位是「塊」。
- 驗證：排序元組-createIndex 正在對索引掃描階段的輸出進行排序。
- 驗證：掃描集合-createIndex 正在掃描集合以驗證前兩個階段中找到的索引鍵。測量單位是「塊」。

索引建置輸出範例

在下面的輸出示例（前景索引構建）中，顯示了索引創建的狀態。「msg」欄位會指出組建的階段和完成百分比，來摘要建置進度。「Worker」欄位會指出在索引建置的該階段期間使用的 Worker 數目。「進度」欄位會顯示用來計算完成百分比的實際數字。

Note

Amazon DocumentDB 4.0 版不支持「currentIndexBuild名稱」，「消息」和「進度」字段。

```
{
  "inprog" : [{
    ...
    "command": {
      "createIndexes": "test",
      "indexes": [{
        "v": 2,
        "key": {
          "user_name": 1
        },
        "name": "user_name_1"
      }],
      "lsid": {
```

```
      "id": UUID("094d0fba-8f41-4373-82c3-7c4c7b5ff13b")
    },
    "$db": "test"
  },
  "currentIndexBuildName": user_name_1,
  "msg": "Index Build: building index number_1, stage 6/6 building index:
656860/1003520 (keys) 65%",
  "workers": 1,
  "progress": {
    "done": 656861,
    "total": 1003520
  },
  ...
],

"ok" : 1
}
```

管理集合層級文件壓縮

Amazon DocumentDB 集合級文件壓縮可讓您壓縮集合中的文件，以降低儲存和 IO 成本。您可以在收集層級啟用文件壓縮，並視需要檢視壓縮指標，方法是透過壓縮指標 (例如壓縮文件的儲存大小和壓縮狀態) 來測量儲存增益。亞馬遜文檔數據庫使用 LZ4 壓縮算法來壓縮文檔。

指導方針

下列準則適用於集合層級文件壓縮：

- 文件壓縮預設為停用
- 文件壓縮無法套用至現有的集合。
- 只有 Amazon 文件資料庫 5.0 版及更新版本才支援文件壓縮。
- 亞馬遜文檔數據庫僅壓縮大小為 2KB 或更大的文檔。

啟用文件壓縮

使用方法在亞馬遜 DocumentDB 上創建集合時啟用 `db.createCollection()` 文檔壓縮：

```
db.createCollection( sample_collection, {
```

```
storageEngine : {
  documentDB: {
    compression:{
      enable: <true | false>
    }
  }
}
})
```

監控文件壓縮

您可以檢查集合是否被壓縮，並按如下方式計算其壓縮比。

從 mongo 殼層執行 `db.printCollectionStats()` 或命 `db.collection.stats()` 令來檢視壓縮統計資料。輸出會顯示原始大小和壓縮大小，您可以比較這些大小以分析文件壓縮所帶來的儲存增益。在此範例中，會顯示名為「sample_collection」之集合的統計資料：

```
db.sample_collection.stats(1024*1024)

{
  "ns" : "test.sample_collection",
  "count" : 1000000,
  "size" : 3906.3,
  "avgObjSize" : 4096,
  "storageSize" : 1953.1,
  compression:{
    "enabled" : true,
    "threshold" : 2032
  }
  ...
}
```

- size-文件集合的原始大小。
- avgObjSize-壓縮前的平均文件大小四捨五入為第一個小數。度量單位為位元組。
- 儲存空間大小-壓縮後的集合儲存大小。度量單位為位元組。
- 啟用-指示壓縮是否已啟用或停用。

若要計算實際的壓縮率，請將集合大小除以儲存大小 (大小/儲存空間大小)。在上述範例中，計算方式為 $3906.3/1953.1$ ，其轉換為 2:1 的壓縮比率。

管理現有的集合

雖然您無法壓縮現有的集合，但您可以轉換未壓縮或壓縮的文件。若要以壓縮格式儲存現有的未壓縮文件，請將文件複製到啟用壓縮的集合中。若要將壓縮文件轉換為未壓縮格式，請將文件複製到已停用壓縮的集合中。

Amazon DocumentDB 事件範例事件管理

Amazon DocumentDB (具有 MongoDB 相容性) 會記錄與您的執行個體、安全群組和執行個體相關的事件。該資訊會包含事件的日期和時間、事件的來源名稱和來源類型，以及與事件相關的訊息。

Important

對於某些管理功能，Amazon DocumentDB 會使用與 Amazon RDS 和亞馬 Amazon Neptune 共用的操作技術。區域限制 (在區域層級管理的限制) 是在亞馬遜文件資料庫、亞馬遜 RDS 和亞馬 Amazon Neptune 之間共用的。如需詳細資訊，請參閱[區域配額](#)。

主題

- [檢視 Amazon DocumentDB DocumentDB 事件類別](#)
- [檢視 Amazon DocumentDB 事件事件範例](#)

檢視 Amazon DocumentDB DocumentDB 事件類別

每個 Amazon DocumentDB 資源類型都有可以與其關聯的特定事件類型。您可以使用此 AWS CLI `describe-event-categories` 操作來檢視事件類型和 Amazon DocumentDB 資源類型之間的對應。

參數

- **--source-type** 選擇性。使用 `--source-type` 參數查看特定來源類型的事件類別。以下是允許值：
 - `db-cluster`
 - `db-instance`
 - `db-parameter-group`
 - `db-security-group`

- `db-cluster-snapshot`
- `--filters` 選擇性。要查看僅 Amazon DocumentDB 的事件類別，請使用過濾器 `--filter Name=engine,Values=docdb`。

Example

以下程式碼會列出與叢集關聯的事件類別。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-event-categories \  
  --filter Name=engine,Values=docdb \  
  --source-type db-cluster
```

針對 Windows：

```
aws docdb describe-event-categories ^  
  --filter Name=engine,Values=docdb ^  
  --source-type db-cluster
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "EventCategoriesMapList": [  
    {  
      "EventCategories": [  
        "notification",  
        "failure",  
        "maintenance",  
        "failover"  
      ],  
      "SourceType": "db-cluster"  
    }  
  ]  
}
```

下列程式碼會列出與每個 Amazon DocumentDB 來源類型相關聯的事件類別。

```
aws docdb describe-event-categories
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "EventCategoriesMapList": [
    {
      "SourceType": "db-instance",
      "EventCategories": [
        "notification",
        "failure",
        "creation",
        "maintenance",
        "deletion",
        "recovery",
        "restoration",
        "configuration change",
        "read replica",
        "backtrack",
        "low storage",
        "backup",
        "availability",
        "failover"
      ]
    },
    {
      "SourceType": "db-security-group",
      "EventCategories": [
        "configuration change",
        "failure"
      ]
    },
    {
      "SourceType": "db-parameter-group",
      "EventCategories": [
        "configuration change"
      ]
    },
    {
      "SourceType": "db-cluster",
      "EventCategories": [
        "notification",
        "failure",
        "maintenance",
        "failover"
      ]
    }
  ]
}
```

```

    },
    {
      "SourceType": "db-cluster-snapshot",
      "EventCategories": [
        "backup"
      ]
    }
  ]
}

```

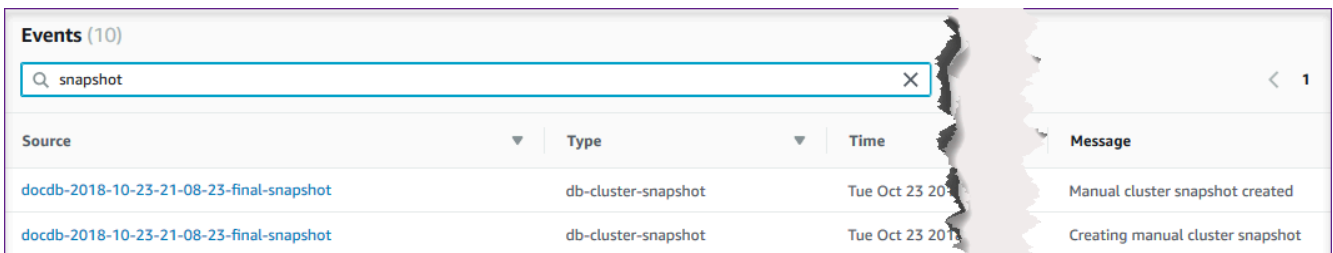
檢視 Amazon DocumentDB 事件事件範例

您可透過 Amazon DocumentDB 事件範例所發生事件的主控制台來擷取 Amazon DocumentDB 資源的事件的事件範例。此外，您還可以使用 [describeDB 操作來擷取 DescribeEventsAmazon DocumentDB 資源的事](#) AWS CLI 件。若您是使用 AWS CLI 或 Amazon DocumentDB API 檢視事件的事件範例，則最多可擷取過去 14 天內的事件。

Using the AWS Management Console

檢視過去 24 小時內的所有 Amazon DocumentDB 執行個體事件範例項事件範例

1. 登入 AWS Management Console，並在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇 Events (事件)。可用的事件隨即會出現在清單中。
3. 使用 Filter (篩選條件) 清單可依類型篩選事件。在文字方塊中輸入期限，以進一步篩選結果。例如，以下螢幕擷取畫面所顯示的所有 Amazon DocumentDB 事件範例所顯示的所有 Amazon DocumentDB



| Source | Type | Time | Message |
|--|---------------------|-----------------|----------------------------------|
| docdb-2018-10-23-21-08-23-final-snapshot | db-cluster-snapshot | Tue Oct 23 2018 | Manual cluster snapshot created |
| docdb-2018-10-23-21-08-23-final-snapshot | db-cluster-snapshot | Tue Oct 23 2018 | Creating manual cluster snapshot |

Using the AWS CLI

檢視過去 7 天內的所有 Amazon DocumentDB 執行個體事件範例項目事件範例

如果要檢視過去 7 天內的所有 Amazon DocumentDB 執行個體事件，請執行 [describeDB 執行個體事件](#) AWS CLI 操作來檢視過去 7--duration 天內的所有 Amazon DocumentDB 執行個體事件。

```
aws docdb describe-events --duration 10080
```

過濾 Amazon DocumentDB 事件

若要查看特定的 Amazon DocumentDB 事件，請搭配下列參數使用此 describe-events 作業。

參數

- **--filter** 必須將傳回的值限制為 Amazon DocumentDB 事件。
用 **Name=engine, Values=docdb** 來篩選僅適用於 Amazon DocumentDB 的所有事件。
- **--source-identifier** 選擇性。要傳回其事件的事件來源識別碼。如果省略，則會傳回所有來源的事件。
- **--source-type** 選用，除非提供 **--source-identifier** 供，否則為必要。如果提供 **--source-identifier**，則 **--source-type** 必須和 **--source-identifier** 的類型一致。以下是允許值：
 - db-cluster
 - db-instance
 - db-parameter-group
 - db-security-group
 - db-cluster-snapshot

以下範例列出您的所有 Amazon DocumentDB 事件範例範例範例範例

```
aws docdb describe-events --filters Name=engine,Values=docdb
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "Events": [
    {
      "SourceArn": "arn:aws:rds:us-east-1:123SAMPLE012:db:sample-cluster-instance3",
```

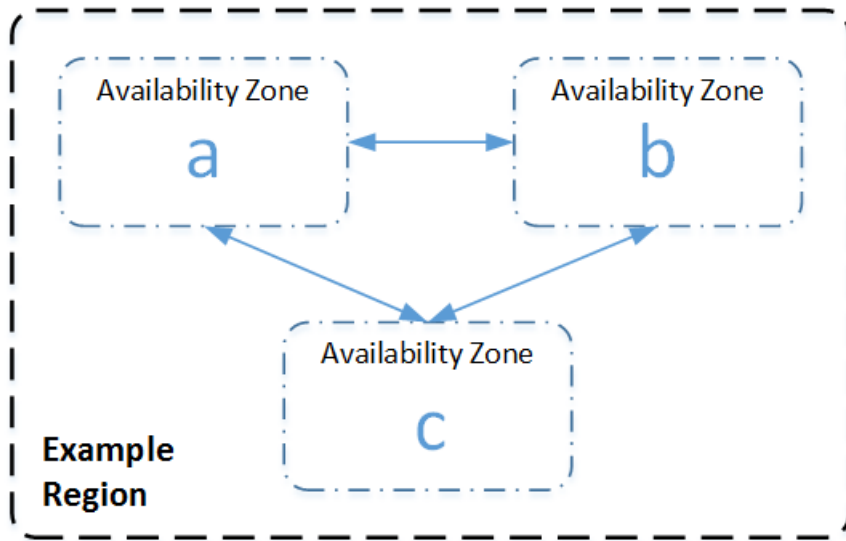
```
    "Message": "instance created",
    "SourceType": "db-instance",
    "Date": "2018-12-11T21:17:40.023Z",
    "SourceIdentifier": "sample-cluster-instance3",
    "EventCategories": [
      "creation"
    ]
  },
  {
    "SourceArn": "arn:aws:rds:us-
east-1:123SAMPLE012:db:docdb-2018-12-11-21-08-23",
    "Message": "instance shutdown",
    "SourceType": "db-instance",
    "Date": "2018-12-11T21:25:01.245Z",
    "SourceIdentifier": "docdb-2018-12-11-21-08-23",
    "EventCategories": [
      "availability"
    ]
  },
  {
    "SourceArn": "arn:aws:rds:us-
east-1:123SAMPLE012:db:docdb-2018-12-11-21-08-23",
    "Message": "instance restarted",
    "SourceType": "db-instance",
    "Date": "2018-12-11T21:25:11.441Z",
    "SourceIdentifier": "docdb-2018-12-11-21-08-23",
    "EventCategories": [
      "availability"
    ]
  }
]
```

如需詳細資訊，請參閱 [稽核 Amazon DocumentDB 事件](#)。

選擇區域與可用區域

Amazon 雲端運算資源託管於全球的多個地點。這些位置包含 AWS 區域 和可用區域。每個 AWS 區域是單獨的地理區域。每個區域擁有多個隔離位置，稱為可用區域。Amazon DocumentDB 讓您能夠將資源 (例如執行個體和資料) 放置在多個位置。除非您特別這樣做，AWS 區域 否則不會複製資源。

Amazon 營運的進階、高度可用的資料中心。儘管故障極為少見，但仍可能影響相同位置內執行個體的可用性。若您將所有執行個體都託管於單一位置，一旦該位置受故障影響，所有執行個體都將無法使用。下圖顯示 AWS 區域 具有三個可用區域。



請務必記住，每個區域都是獨立的。您啟動的任何 Amazon DocumentDB 活動 (例如，建立執行個體或列出可用執行個體) 只會以目前的預設值執行。AWS 區域您可以在主控台上設定 `EC2_REGION` 環境變數來變更預設區域。或者，您可以使用 AWS CLI 中的 `--region` 參數來覆寫它。如需詳細資訊，請參閱〈[配置環境變數和命令行選項](#)〉一節。AWS Command Line Interface

當您使用 Amazon DocumentDB 主控台建立叢集，並選擇在不同的可用區域中建立複本時，Amazon DocumentDB 會建立兩個執行個體。它會在某一個可用區域中建立主要執行個體，並在另一個可用區域中建立複本執行個體。叢集磁碟區一律複寫到三個可用區域。

若要建立或使用特定的 Amazon DocumentDB 執行個體 AWS 區域，請使用對應的區域服務端點。

區域可用性

Amazon DocumentDB 在以下 AWS 區域提供。

Amazon DocumentDB 支援的區域

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------|-----------|-----------|
| 美國東部 (俄亥俄) | us-east-2 | 3 |
| 美國東部 (維吉尼亞北部) | us-east-1 | 6 |

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------------|----------------|-----------|
| 美國西部 (奧勒岡) | us-west-2 | 4 |
| 南美洲 (聖保羅) | sa-east-1 | 3 |
| 亞太區域 (香港) | ap-east-1 | 3 |
| 亞太區域 (海德拉巴) | ap-south-2 | 3 |
| 亞太區域 (孟買) | ap-south-1 | 3 |
| 亞太區域 (首爾) | ap-northeast-2 | 4 |
| 亞太區域 (新加坡) | ap-southeast-1 | 3 |
| 亞太區域 (悉尼) | ap-southeast-2 | 3 |
| 亞太區域 (東京) | ap-northeast-1 | 3 |
| 加拿大 (中部) | ca-central-1 | 3 |
| 中國 (北京) 區域 | cn-north-1 | 3 |
| 中國 (寧夏) | cn-northwest-1 | 3 |
| 歐洲 (法蘭克福) | eu-central-1 | 3 |
| 歐洲 (愛爾蘭) | eu-west-1 | 3 |
| 歐洲 (倫敦) | eu-west-2 | 3 |
| 歐洲 (米蘭) | eu-south-1 | 3 |
| Europe (Paris) | eu-west-3 | 3 |
| 中東 (阿拉伯聯合大公國) | me-central-1 | 3 |
| AWS GovCloud (美國西部) | us-gov-west-1 | 3 |

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------------|---------------|-----------|
| AWS GovCloud (美國東部) | us-gov-east-1 | 3 |

根據預設，Amazon DocumentDB 叢集的時區為世界標準時間 (UTC)。

如需在特定區域尋找叢集和執行個體連線端點的資訊，請參閱[了解亞馬遜文檔數據庫端點](#)。

管理 Amazon DocumentDB 叢集參數群組

您可以使用叢集參數群組中的參數來管理 Amazon DocumentDB 引擎組態。叢集參數群組是 Amazon DocumentDB 組態值的集合，可讓您更輕鬆地管理 Amazon DocumentDB 叢集的參數。叢集參數群組扮演引擎組態值的容器，以套用至資料庫叢集中的所有執行個體。

本節說明如何建立、檢視和修改叢集參數群組。此外也會為您示範如何判斷有哪些叢集參數群組與指定的叢集相關。

主題

- [描述 Amazon DocumentDB 庫客製參數群組](#)
- [建立 Amazon DocumentDB 叢集參數群組](#)
- [修改 Amazon DocumentDB 叢集參數群組](#)
- [修改 Amazon DocumentDB 叢集以使用自訂叢集參數群組](#)
- [複製 Amazon DocumentDB 叢集參數群組](#)
- [重設 Amazon DocumentDB 叢集參數群組](#)
- [刪除 Amazon DocumentDB 叢集參數群組](#)
- [Amazon DocumentDB 叢集參數參考](#)

描述 Amazon DocumentDB 庫客製參數群組

當您在新區域中建立第一個 default Amazon DocumentDB 叢集或使用新引擎時，會自動建立叢集參數群組。在相同區域中建立且具有相同引擎版本的後續叢集會使用 default 叢集參數群組建立。

主題

- [描述 Amazon DocumentDB 集群參數組的詳細信息](#)
- [確定 Amazon DocumentDB 集群的參數組](#)

描述 Amazon DocumentDB 集群參數組的詳細信息

如果要說明指定叢集參數群組的詳細資訊，請使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 完成以下步驟。

Using the AWS Management Console

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 Cluster parameter groups (叢集參數群組) 窗格中，選取您想要查看詳細資訊的參數群組名稱。
4. 產生的頁面會顯示參數群組的參數、最近的活動和標籤。
 - 在 Cluster parameters (叢集參數) 下，您可以看到參數名稱、目前值、允許值、參數是否可修改、套用類型、資料類型和說明。您可以選取參數，然後在叢集參數段落中選擇編輯來修改個別參數。如需詳細資訊，請參閱 [修改 Amazon DocumentDB 叢集參數](#)。
 - 在 Recent events (最近的事件) 中，您可以查看此參數群組的最近事件。您可以使用本節中的搜尋列來篩選這些事件。如需詳細資訊，請參閱 [Amazon DocumentDB 事件範例事件管理](#)。
 - 在 Tags (標籤)，您可以查看在此叢集參數群組上的標籤。您可以在「標籤」區段中選擇「編輯」來新增或移除標籤。如需詳細資訊，請參閱 [標記亞馬遜文檔數據庫資源](#)。

Using the AWS CLI

您可以使用此 `describe-db-cluster-parameter-groups` AWS CLI 命令來檢視 Amazon 資源名稱 (ARN)、系列、說明以及單一叢集參數群組的名稱，或您擁有的 Amazon DocumentDB 所有叢

集參數群組的名稱。您也可以使用指 `describe-db-cluster-parameters` AWS CLI 命令來檢視參數，以及它們在單一叢集參數群組中的詳細資訊。

- **`--describe-db-cluster-parameter-groups`**— 查看所有叢集參數群組及其詳細資料的清單。
 - **`--db-cluster-parameter-group-name`**-可選。您想要說明的叢集參數群組名稱。如果省略這個參數，所有叢集參數群組會加上說明。
- **`--describe-db-cluster-parameters`**— 列出參數群組內的所有參數及其值。
 - **`--db-cluster-parameter-group name`** – 必要。您想要說明的叢集參數群組名稱。

Example

以下程式碼列出高達 100 個叢集參數群組及其 ARN、系列、說明和名稱。

```
aws docdb describe-db-cluster-parameter-groups
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "DBClusterParameterGroups": [
    {
      "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:012345678912:cluster-pg:default.docdb4.0",
      "DBParameterGroupFamily": "docdb4.0",
      "Description": "Default cluster parameter group for docdb4.0",
      "DBClusterParameterGroupName": "default.docdb4.0"
    },
    {
      "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:012345678912:cluster-pg:sample-parameter-group",
      "DBParameterGroupFamily": "docdb4.0",
      "Description": "Custom docdb4.0 parameter group",
      "DBClusterParameterGroupName": "sample-parameter-group"
    }
  ]
}
```

Example

以下程式碼列出 `sample-parameter-group` 的 ARN、系列、說明和名稱。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameter-groups \  
  --db-cluster-parameter-group-name sample-parameter-group
```

針對 Windows：

```
aws docdb describe-db-cluster-parameter-groups ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBClusterParameterGroups": [  
    {  
      "DBClusterParameterGroupArn": "arn:aws:rds:us-  
east-1:123456789012:cluster-pg:sample-parameter-group",  
      "Description": "Custom docdb4.0 parameter group",  
      "DBParameterGroupFamily": "docdb4.0",  
      "DBClusterParameterGroupName": "sample-parameter-group"  
    }  
  ]  
}
```

Example

以下程式碼列出 `sample-parameter-group` 中參數的值。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name sample-parameter-group
```

針對 Windows：

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```


此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "Parameters": [
    {
      "ParameterName": "audit_logs",
      "ParameterValue": "disabled",
      "Description": "Enables auditing on cluster.",
      "Source": "system",
      "ApplyType": "dynamic",
      "DataType": "string",
      "AllowedValues": "enabled,disabled",
      "IsModifiable": true,
      "ApplyMethod": "pending-reboot"
    },
    {
      "ParameterName": "change_stream_log_retention_duration",
      "ParameterValue": "17777",
      "Description": "Duration of time in seconds that the change stream log
is retained and can be consumed.",
      "Source": "user",
      "ApplyType": "dynamic",
      "DataType": "integer",
      "AllowedValues": "3600-86400",
      "IsModifiable": true,
      "ApplyMethod": "pending-reboot"
    }
  ]
}
```

確定 Amazon DocumentDB 集群的參數組

如果要判斷有哪些參數群組與特定叢集相關，請使用 AWS Management Console 或 AWS CLI 完成以下步驟。

Using the AWS Management Console

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在左側導覽窗格中選擇 Clusters (叢集)。
3. 在叢集的清單中，選取您有興趣之叢集的名稱。

4. 產生的頁面會顯示您所選取叢集的詳細資訊。向下捲動至 Cluster details (叢集詳細資訊)。在該區段底部的 Cluster parameter group (叢集參數群組) 下找到參數群組的名稱。

Cluster details

Configurations and status

ARN

arn:aws:rds: :cluster:sample-cluster

Cluster identifier

sample-cluster (available)

Cluster creation time

1/10/2020, 2:13:38 PM UTC-8

Cluster endpoint

sample-cluster. .docdb.amazonaws.com

Reader endpoint

sample-cluster. .docdb.amazonaws.com

Master username

Port

27017

Status

available

Cluster parameter group

sample-parameter-group

Deletion protection

Enabled

CloudWatch logs enabled

None

Using the AWS CLI

以下 AWS CLI 程式碼會判斷哪些參數群組管理叢集 `sample-cluster`。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  [  
    "sample-cluster",  
    "sample-parameter-group"  
  ]  
]
```

建立 Amazon DocumentDB 叢集參數群組

當您使用新引擎版本和新區域建立叢集時 `default.docdb3.6`，會建立預設叢集參數群組，例如、或 `default.docdb5.0` `default.docdb4.0` 在此區域中建立且具有相同引擎版本的後續叢集會繼承 `default` 叢集參數群組。建立 `default` 參數群組後，就無法刪除或更名參數群組。您可以建立具有偏好參數值的自訂參數群組，並將其附加到 Amazon DocumentDB 叢集，以修改叢集執行個體的引擎行為。

下列程序會引導您完成建立自訂叢集參數群組。您接著可 [在該參數群組中修改參數](#)。


Note

建立叢集參數群組後，您應等待至少 5 分鐘，然後再使用該特定參數群組。這可讓 Amazon DocumentDB 在將叢集參數群組用於新叢集之前完成 `create` 動作。您可以使用 AWS Management Console 或 `describe-db-cluster-parameter-groups` AWS CLI 操作，以驗證您的叢集參數群組是否已建立。如需詳細資訊，請參閱 [描述 Amazon DocumentDB 庫客製參數群組](#)。

Using the AWS Management Console

建立叢集參數群組

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

 Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 Cluster parameter groups (叢集參數群組) 窗格中，選擇 Create (建立)。
4. 在 Create cluster parameter group (建立叢集參數群組) 窗格中，輸入以下項目：
 - a. 群組名稱 — 輸入叢集參數群組的名稱。例如 `sample-parameter-group`。叢集參數群組具有以下命名限制：
 - 長度為 [1-255] 個英數字元。
 - 第一個字元必須是字母。
 - 不能以連字號結尾或連續包含兩個連字號。
 - b. 說明 — 提供此叢集參數群組的說明。
5. 若要建立叢集參數群組，請選擇 Create (建立)。若要取消操作，請選擇 Cancel (取消)。
6. 選擇 Create (建立) 之後，頁面頂端會顯示下列文字，以確認您的叢集參數群組是否已成功建立：

Successfully created cluster parameter group '*sample-parameter-group*'.

Using the AWS CLI

若要為 Amazon DocumentDB 4.0 叢集建立新的叢集參數群組，請使用具有下列參數的 AWS CLI `create-db-cluster-parameter-group` 作業：

- **--db-cluster-parameter-group-name** — 自訂叢集參數群組的名稱。例如 `sample-parameter-group`。

- **--db-cluster-parameter-group-family**— 用作自訂叢集參數群組範本的叢集參數群組族群。目前，此項目必須是 docdb4.0。
- **--description**— 此叢集參數群組的使用者提供的說明。以下範例使用 "Custom docdb4.0 parameter group"。

若為 Linux、macOS 或 Unix：

Example

```
aws docdb create-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group \  
  --db-parameter-group-family docdb4.0 \  
  --description "Custom docdb4.0 parameter group"
```

針對 Windows：

```
aws docdb create-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name sample-parameter-group ^  
  --db-parameter-group-family docdb4.0 ^  
  --description "Custom docdb4.0 parameter group"
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBClusterParameterGroup": {  
    "DBClusterParameterGroupName": "sample-parameter-group",  
    "DBParameterGroupFamily": "docdb4.0",  
    "Description": "Custom docdb4.0 parameter group",  
    "DBClusterParameterGroupArn": "sample-parameter-group-arn"  
  }  
}
```

修改 Amazon DocumentDB 叢集參數群組

本節說明如何修改自訂的 Amazon DocumentDB 參數群組。在 Amazon DocumentDB 中，您無法修改在新區 default 域中第一次使用新引擎版本建立叢集時所建立的叢集參數群組。如果 Amazon DocumentDB 叢集使用預設叢集參數群組，而您想要修改其中的值，則必須先[建立新參數群組或複製現有參數群組](#)、修改它，然後將修改後的參數群組套用至叢集。

若要修改自訂叢集參數群組，請完成下列步驟。修改動作可能需要一段時間才能傳播。請等待已修改的叢集參數群組可用，然後再將它附加到叢集。您可以使用 AWS Management Console 或 AWS CLI `describe-db-cluster-parameters` 操作，以驗證您的叢集參數群組是否已修改。如需詳細資訊，請參閱[描述叢集參數群組](#)。

Using the AWS Management Console

請依照下列步驟修改自訂的 Amazon DocumentDB 參數群組。您無法修改 `default` 參數群組。如果您想要修改 `default` 參數群組中的值，可以[複製預設叢集參數群組](#)並加以修改，接著將修改的參數群組套用到叢集。如需將參數群組套用到叢集的詳細資訊，請參閱[修改 Amazon DocumentDB 叢集](#)。

修改自訂叢集參數群組

1. [登入 AWS Management Console](#)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在主控台左側的導覽窗格中，選擇 Parameter groups (參數群組)。在參數群組清單中，選擇您要修改看的參數群組名稱。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。


3. 對於您要修改之參數群組中的每個參數，請執行下列動作：
 - a. 找出您要修改的參數，並檢查「可修改」欄 `true` 下是否列示該參數，以確認該參數是否可修改。
 - b. 如果可以修改，請選取參數並從主控台頁面的右上角選擇 Edit (編輯)。
 - c. 在修改 `<parameter-name>` 對話方塊中，進行所需的變更。然後，選擇 Modify cluster parameter (修改叢集參數)，或選擇 Cancel (取消) 以捨棄變更。

Using the AWS CLI

您可以使用 `ParameterValue` 修改自訂 Amazon DocumentDB 叢集參數群組中任何可修改參數 `ApplyMethod` 的 `Description`、或。AWS CLI 您無法直接修改預設叢集參數群組。

若要修改自訂叢集參數群組的參數，請使用 `modify-db-cluster-parameter-group` 操作搭配下列參數。

- **--db-cluster-parameter-group-name** – 必要。您想要修改的叢集參數群組名稱。
- **--parameters** – 必要。您要修改的參數。如需適用於 Amazon DocumentDB 叢集中所有執行個體的參數清單，請參閱 [Amazon DocumentDB 群集參數參考](#) 每個參數項目必須包括下列：
 - **ParameterName**— 您要修改的參數名稱。
 - **ParameterValue**— 此參數的新值。
 - **ApplyMethod**— 套用此參數變更的方式。允許值為 `immediate` 和 `pending-reboot`。

 Note

`static` 必須擁有使用 `ApplyType` 的參數 `ApplyMethod pending-reboot`。

Example - 修改參數值

在此範例中，您列出 `sample-parameter-group` 的參數值和修改 `tls` 參數。接著，在等待 5 分鐘後，您會再次列出 `sample-parameter-group` 的參數值，以查看變更的參數值。

1. 列出參數和其 `sample-parameter-group` 的值。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name sample-parameter-group
```

針對 Windows：

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "Parameters": [  
    {  
      "Source": "system",  
      "ApplyType": "static",  
      "AllowedValues": "disabled,enabled",  
      "ParameterValue": "enabled",  
      "ApplyMethod": "pending-reboot",
```



```

        "DataType": "string",
        "ParameterName": "tls",
        "IsModifiable": true,
        "Description": "Config to enable/disable TLS"
    },
    {
        "Source": "user",
        "ApplyType": "dynamic",
        "AllowedValues": "disabled,enabled",
        "ParameterValue": "enabled",
        "ApplyMethod": "pending-reboot",
        "DataType": "string",
        "ParameterName": "ttl_monitor",
        "IsModifiable": true,
        "Description": "Enables TTL Monitoring"
    }
]
}

```

2. 修改 `tls` 參數，讓其值為 `disabled`。

您無法修改 `ApplyMethod`，因為 `ApplyType` 是 `static`。

若為 Linux、macOS 或 Unix：

```

aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
    "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot

```

針對 Windows：

```

aws docdb modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name sample-parameter-group ^
    --parameters
    "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot

```

此操作的輸出將會如下所示 (JSON 格式)。

```

{
    "DBClusterParameterGroupName": "sample-parameter-group"
}

```

3. 等待至少 5 分鐘。
4. 列出 `sample-parameter-group` 的參數值，來確認 `tls` 參數已經過修改。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name sample-parameter-group
```

針對 Windows：

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "Parameters": [  
    {  
      "ParameterValue": "false",  
      "ParameterName": "enable_audit_logs",  
      "ApplyType": "dynamic",  
      "DataType": "string",  
      "Description": "Enables auditing on cluster.",  
      "AllowedValues": "true,false",  
      "Source": "system",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot"  
    },  
    {  
      "ParameterValue": "disabled",  
      "ParameterName": "tls",  
      "ApplyType": "static",  
      "DataType": "string",  
      "Description": "Config to enable/disable TLS",  
      "AllowedValues": "disabled,enabled",  
      "Source": "system",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot"  
    }  
  ]  
}
```

修改 Amazon DocumentDB 叢集以使用自訂叢集參數群組

當您建立 Amazon DocumentDB 叢集時，會自動為該叢集建立一個 default.docdb4.0 參數群組。您無法修改 default 叢集參數群組。您可以改為修改 Amazon DocumentDB 叢集，將新的自訂參數群組與叢集建立關聯。

本節說明如何使用 AWS Management Console 和 AWS Command Line Interface (AWS CLI) 修改現有的 Amazon DocumentDB 叢集以使用自訂叢集參數群組。

Using the AWS Management Console

修改 Amazon DocumentDB 叢集以使用新的非預設叢集參數群組

1. 在開始之前，請確定您已建立 Amazon DocumentDB 叢集和叢集參數群組。如需進一步說明，請參閱 [創建一個 Amazon DocumentDB 叢集](#) 與 [建立 Amazon DocumentDB 叢集參數群組](#)。
2. 建立叢集參數群組後，請在的位置開啟 Amazon DocumentDB 主控台。 <https://console.aws.amazon.com/docdb> 在導覽窗格中選擇 Clusters (叢集) 來將新的參數群組新增至叢集。
3. 選擇您要與參數群組建立關聯的叢集。選擇 Actions (動作)，然後選擇 Modify (修改) 來修改叢集。
4. 在 Cluster options (叢集選項) 下，選擇要與叢集建立關聯的參數群組。
5. 選擇 Continue (繼續) 以檢視修改摘要。
6. 確認變更後，即可立即套用，或是在下一個維護時段在 Scheduling of modifications (修改的排程) 下套用。
7. 選擇 Modify cluster (修改叢集) 來以新的參數群組更新叢集。

Using the AWS CLI

在開始之前，請確定您已建立 Amazon DocumentDB 叢集和叢集參數群組。您可以使用此 [AWS CLI create-db-cluster](#) 操作 [建立 Amazon DocumentDB 叢集](#)。您可以使用此 [AWS CLI create-db-cluster-parameter-group](#) 作業 [建立叢集參數群組](#)。

如果要將新的叢集參數群組新增至叢集，請使用 AWS CLI `modify-db-cluster` 操作搭配下列參數。

- `--db-cluster-identifier` — 叢集的名稱 (例如，`sample-cluster`)。
- `--db-cluster-parameter-group-name` — 您要與叢集建立關聯的參數群組名稱 (例如 `sample-parameter-group`)。

Example

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的輸出將會如下所示 (JSON 格式)。

```
"DBCluster": {  
  "AvailabilityZones": [  
    "us-west-2c",  
    "us-west-2b",  
    "us-west-2a"  
  ],  
  "BackupRetentionPeriod": 1,  
  "DBClusterIdentifier": "sample-cluster",  
  "DBClusterParameterGroup": "sample-parameter-group",  
  "DBSubnetGroup": "default",  
  ...  
}
```

複製 Amazon DocumentDB 叢集參數群組

您可以使用AWS Management Console或 AWS Command Line Interface (AWS CLI) 在 Amazon DocumentDB 中製作叢集參數群組的副本。

Using the AWS Management Console

下列程序會藉由製作現有叢集參數群組的複本，引導您建立新的叢集參數群組。

複製叢集參數群組

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在 Cluster parameter groups (叢集參數群組) 窗格中，選擇您想要複製的叢集參數群組名稱。
4. 依序選擇 Actions (動作) 和 Copy (複製) 來複製該參數群組。
5. 在 Copy options (複製選項) 下，輸入新叢集參數群組的名稱和說明。接著選擇 Copy (複製) 來儲存變更。

Using the AWS CLI

若要複製叢集參數群組，請使用 `copy-db-cluster-parameter-group` 操作搭配下列參數。

- **--source-db-cluster-parameter-group-identifier** – 必要。您要複製的叢集參數群組名稱或 Amazon Resource Name (ARN)。

如果來源和目標叢集參數群組位於相同AWS 區域，則識別碼可以是名稱或 ARN。

如果來源和目標叢集參數群組不同AWS 區域，則識別碼必須是 ARN。

- **--target-db-cluster-parameter-group-identifier** – 必要。叢集參數群組複本的名稱或 ARN。

約束:

- 不得為無效、空白或留空。
 - 必須包含 1—255 個字母、數字或連字號。
 - 第一個字元必須是字母。
 - 不能以連字號結尾或連續包含兩個連字號。
- **--target-db-cluster-parameter-group-description** – 必要。使用者提供叢集參數群組複本的說明。

Example

以下程式碼複製 `sample-parameter-group`，將複本命名為 `sample-parameter-group-copy`。

若為 Linux、macOS 或 Unix：

```
aws docdb copy-db-cluster-parameter-group \  
  --source-db-cluster-parameter-group-identifier sample-parameter-group \  
  --target-db-cluster-parameter-group-identifier sample-parameter-group-copy \  
  --target-db-cluster-parameter-group-description "Copy of sample-parameter-group"
```

針對 Windows：

```
aws docdb copy-db-cluster-parameter-group ^  
  --source-db-cluster-parameter-group-identifier sample-parameter-group ^  
  --target-db-cluster-parameter-group-identifier sample-parameter-group-copy ^
```

```
--target-db-cluster-parameter-group-description "Copy of sample-parameter-group"
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "DBClusterParameterGroup": {
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-pg:sample-parameter-group-copy",
    "DBClusterParameterGroupName": "sample-parameter-group-copy",
    "DBParameterGroupFamily": "docdb4.0",
    "Description": "Copy of sample-parameter-group"
  }
}
```

重設 Amazon DocumentDB 叢集參數群組

您可以使用或 AWS Command Line Interface (AWS CLI) 重設叢集參數群組，將部分AWS Management Console或全部 Amazon DocumentDB 叢集參數群組的參數值重設為預設值。

Using the AWS Management Console

請遵循這些步驟操作，將部分或全部叢集參數群組的參數值重設為預設值。

重設叢集參數群組的參數值

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在主控台左側的導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在 Cluster parameter groups (叢集參數群組) 窗格中，選擇您想要重設的叢集參數群組名稱。
4. 依序選擇 Actions (動作) 和 Reset (重設) 來重設該參數群組。
5. 在產生的 Cluster parameter group reset confirmation (叢集參數群組重設確認) 頁面上，確認您想要將該參數群組的所有叢集參數重設為預設值。然後選擇 Reset (重設) 以重設您的參數群組。您也可以選擇 Cancel (取消) 以捨棄變更。

Using the AWS CLI

若要將部分或全部叢集參數群組的參數值重設為預設值，使用 `reset-db-cluster-parameter-group` 操作搭配下列參數。

- **--db-cluster-parameter-group-name** – 必要。要重設之叢集參數群組的名稱。
- **--parameters**-可選。叢集參數群組中要重設為預設值的 ApplyMethod 和 ParameterName 清單。靜態參數必須設定為 pending-reboot，才會在下一個執行個體重新啟動或 reboot-db-instance 請求時生效。您必須呼叫您希望將更新的靜態參數套用到其中的叢集中 reboot-db-instance 的每個執行個體。

此參數和 --reset-all-parameters 互斥：您可以使用其中一個，但無法同時使用兩者。

- **--reset-all-parameters** 或 **--no-reset-all-parameters**-可選。指定是否要將所有參數 (--reset-all-parameters) 或僅一些參數 (--no-reset-all-parameters) 重設為預設值。--reset-all-parameters 參數和 --parameters 互斥：您可以使用其中一個，但無法同時使用兩者。

重設整個群組時，動態參數會立即更新。靜態參數必須設定為 pending-reboot，才會在下一個執行個體重新啟動或 reboot-db-instance 請求時生效。您必須呼叫您希望將更新的靜態參數套用到其中的叢集中 reboot-db-instance 的每個執行個體。

Example

範例 1：將所有參數重設為預設值

以下程式碼會將在叢集參數群組 sample-parameter-group 中的所有參數重設為預設值。

若為 Linux、macOS 或 Unix：

```
aws docdb reset-db-cluster-parameter-group \  
    --db-cluster-parameter-group-name sample-parameter-group \  
    --reset-all-parameters
```

針對 Windows：

```
aws docdb reset-db-cluster-parameter-group ^  
    --db-cluster-parameter-group-name sample-parameter-group ^  
    --reset-all-parameters
```

範例 2：將指定參數重設為預設值

以下程式碼會將在叢集參數群組 sample-parameter-group 中的 tls 參數重設為預設值。

若為 Linux、macOS 或 Unix：

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group \  
  --no-reset-all-parameters \  
  --parameters ParameterName=tls,ApplyMethod=pending-reboot
```

針對 Windows :

```
aws docdb reset-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name sample-parameter-group ^  
  --no-reset-all-parameters ^  
  --parameters ParameterName=tls,ApplyMethod=pending-reboot
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBClusterParameterGroupName": "sample-parameter-group"  
}
```

重新啟動叢集執行個體

在靜態參數的值變更前，必須將叢集執行個體重新開機。在您希望將更新靜態參數套用到其中的叢集中重新啟動每個執行個體。

若為 Linux、macOS 或 Unix :

```
aws docdb reboot-db-instance \  
  --db-instance-identifier sample-cluster-instance
```

針對 Windows :

```
aws docdb reboot-db-instance ^  
  --db-instance-identifier sample-cluster-instance
```

刪除 Amazon DocumentDB 集群參數組

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 刪除自訂的 Amazon DocumentDB 叢集參數群組。您無法刪除 default.docdb4.0 叢集參數群組。

Using the AWS Management Console

刪除叢集參數群組

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示

(☰

)。

3. 在 Parameter groups (參數群組) 窗格中，選擇您想要刪除的叢集參數群組左側旁的選項按鈕。
4. 選擇動作，然後選擇刪除。
5. 在 Delete (刪除) 確認窗格中，選擇 Delete (刪除) 來刪除叢集參數群組。如果要保留叢集參數群組，請選擇 Cancel (取消)。

Using the AWS CLI

若要刪除叢集參數群組，請使用 `delete-db-cluster-parameter-group` 操作搭配下列參數。

- **`--db-cluster-parameter-group-name`** – 必要。要刪除之叢集參數群組的名稱。這必須是現有叢集的參數群組。您無法刪除 `default.docdb4.0` 叢集參數群組。

Example - 刪除叢集參數群組

以下範例逐步引導您刪除叢集參數群組的三個步驟：

1. 尋找您想要刪除的叢集參數群組名稱。
2. 刪除指定的叢集參數群組。
3. 確認已刪除叢集的參數群組。

1. 尋找您想要刪除的叢集參數群組名稱。

以下程式碼列出所有叢集參數群組的名稱。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameter-groups \  
  --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

針對 Windows :

```
aws docdb describe-db-cluster-parameter-groups ^  
  --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

上述操作的輸出是與以下 (JSON 格式) 類似的叢集參數群組名稱清單。

```
[  
  [  
    "default.docdb4.0"  
  ],  
  [  
    "sample-parameter-group"  
  ],  
  [  
    "sample-parameter-group-copy"  
  ]  
]
```

2. 刪除特定叢集參數群組。

以下程式碼會刪除叢集參數群組 `sample-parameter-group-copy`。

若為 Linux、macOS 或 Unix :

```
aws docdb delete-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group-copy
```

針對 Windows :

```
aws docdb delete-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name sample-parameter-group-copy
```

此操作將不會有輸出。

3. 確認已刪除指定的叢集參數群組。

以下程式碼列出所有剩餘叢集參數群組的名稱。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameter-groups \
  --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

針對 Windows：

```
aws docdb describe-db-cluster-parameter-groups ^
  --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

上述操作的輸出是與以下 (JSON 格式) 類似的叢集參數群組清單。您剛刪除的叢集參數群組不應在清單中。

此操作的輸出將會如下所示 (JSON 格式)。

```
[
  [
    "default.docdb4.0"
  ],
  [
    "sample-parameter-group"
  ]
]
```

Amazon DocumentDB 群集參數參考

當您變更動態參數並儲存叢集參數群組時，無論 Apply Immediately (立即套用) 設定為何，變更都會立即套用。當您變更靜態參數並儲存叢集參數群組時，參數變更會在您手動重新啟動叢集執行個體之後生效。您可以使用 Amazon DocumentDB 主控台或明確呼叫來重新啟動執行個體。reboot-db-instance

下表顯示套用至 Amazon DocumentDB 叢集中所有執行個體的參數。

Amazon DocumentDB 群集級參數

| 參數 | 預設值 | 有效值 | 可修改 | 套用類型 | 資料類型 | 描述 |
|------------|----------|-------------|-----|------|------|-------------------------|
| audit_logs | disabled | 啟用、停用、DDL、讀 | 是 | 動態 | 字串 | 定義是否啟用 Amazon CloudWatc |

| 參數 | 預設值 | 有效值 | 可修改 | 套用類型 | 資料類型 | 描述 |
|----|-----|--------------------------|-----|------|------|--|
| | | 取、dml_w rite、全 部、無 | | | | <p>h 稽核日誌。</p> <ul style="list-style-type: none"> • enabled—已啟用 CloudWatch 稽核記錄。 • disabled—CloudWatch 稽核記錄已停用。 • ddl已啟用 DDL 事件的稽核功能。 • dml_read已啟用 DML 讀取事件的稽核。 • dml_write已啟用 DML 寫入事件的稽核功能。 • all已啟用所有資料庫事件的稽核功能。 |

| 參數 | 預設值 | 有效值 | 可修改 | 套用類型 | 資料類型 | 描述 |
|--------------------------------------|-------|-------------|-----|------|---------|--|
| | | | | | | <ul style="list-style-type: none"> • none—稽核已停用。 |
| change_stream_log_retention_duration | 10800 | 3600-604800 | 是 | 動態 | Integer | 定義變更串流日誌保留並可使用的持續時間 (以秒為單位)。 |
| profiler | 停用 | 啟用, 停用 | 是 | 動態 | 字串 | <p>啟用緩慢操作的分析功能。</p> <ul style="list-style-type: none"> • enabled—超過客戶定義閾值 (例如 100 毫秒) 所花費的操作會記錄到 Amazon 日誌中。CloudWatch • disabled慢速操作不會記錄到 CloudWatch 日誌中。 |

| 參數 | 預設值 | 有效值 | 可修改 | 套用類型 | 資料類型 | 描述 |
|------------------------|-----|---------------|-----|------|---------|---|
| profiler_sampling_rate | 1.0 | 0.0-1.0 | 是 | 動態 | Float | 定義已記錄操作的取樣率。 |
| profiler_threshold_ms | 100 | 50-2147483646 | 是 | 動態 | Integer | <p>定義 profiler 的閾值。</p> <ul style="list-style-type: none"> 所有大於的作業 profiler_threshold_ms 都會記錄至 CloudWatch 記錄檔。 |

| 參數 | 預設值 | 有效值 | 可修改 | 套用類型 | 資料類型 | 描述 |
|-----|-----|------------------|-----|------|------|---|
| tls | 已啟用 | 啟用，禁用，FIPS-140-3 | 是 | 靜態 | 字串 | <p>定義 Transport Layer Security (TLS) 連線是否必要。</p> <ul style="list-style-type: none"> • enabled需要 TLS 連線才能連線。 • disabled無法使用 TLS 連線來連線。 • fips-140-3 必須與聯邦資訊處理標準 (FIPS) 屬性進行 TLS 連線才能連線。叢集僅接受每個 FIPS 發行集 140-3 的安全連線。只有從下列區域中的 Amazon |

| 參數 | 預設值 | 有效值 | 可修改 | 套用類型 | 資料類型 | 描述 |
|-----------------|-----|-------|-----|------|------|--|
| | | | | | | DocumentDB 5.0 (引擎版本 3.0.3727) 叢集開始才支援此功能： CA-中央 -1、美國西部 -2、美國東部 -1、美國東部 -2、-1、-1。 us-gov-east us-gov-west |
| ttl_monit or | 已啟用 | 啟用，停用 | 是 | 動態 | 字串 | 定義是否為叢集啟用存活期 (TTL) 監控功能。 <ul style="list-style-type: none"> • enabled 已啟用 TTL 監視。 • disabled — TTL 監視已停用。 |

修改 Amazon DocumentDB 叢集參數

在 Amazon DocumentDB 中，叢集參數群組包含適用於您在叢集中建立的所有執行個體的參數。如為自訂叢集參數群組，您隨時可以為您建立的參數群組修改其參數值，或將所有參數值重設為預設值。

本節說明如何檢視組成 Amazon DocumentDB 叢集參數群組的參數及其值，以及如何變更或更新這些值。

參數可以是 dynamic (動態) 或 static (靜態)。當您變更動態參數並儲存叢集參數群組時，無論 Apply Immediately 設定為何，變更都會立即套用。當您變更靜態參數並儲存叢集參數群組時，參數變更只會在您手動重新啟動叢集執行個體之後生效。

檢視亞 Amazon DocumentDB 叢集參數群組的參數

您可以使用或查看 Amazon DocumentDB 叢集的參數及其值。AWS Management Console AWS CLI

Using the AWS Management Console

若要檢視叢集參數群組的詳細資訊

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 Parameter groups (參數群組) 窗格中，選擇您想要查看詳細資訊的叢集參數群組名稱。
4. 結果頁面會顯示各個參數的下列值：參數名稱、目前值、允許值、參數是否可修改、來源、資料類型和說明。

| | Cluster parameter name ▲ | Values ▼ | Allowed values |
|-----------------------|--------------------------|----------|------------------|
| <input type="radio"/> | audit_logs | disabled | enabled,disabled |
| <input type="radio"/> | tls | enabled | disabled,enabled |
| <input type="radio"/> | ttl_monitor | enabled | disabled,enabled |

Using the AWS CLI

若要查看叢集參數群組的參數和其值，請使用 describe-db-cluster-parameters 操作搭配下列參數。

- **--db-cluster-parameter-group-name** – 必要。您想要詳細參數清單的叢集參數群組名稱。

- **--source**-可選。如果提供，只會傳回特定來源的參數。參數來源可以是 `engine-default`、`system` 或 `user`。

Example

以下程式碼列出 `custom3-6-param-grp` 參數群組的參數及其值。如需參數群組的詳細資訊，請省略 `--query` 行。如需所有參數群組的詳細資訊，請省略 `--db-cluster-parameter-group-name` 行。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --query 'Parameters[*].[ParameterName,ParameterValue]'
```

針對 Windows：

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name custom3-6-param-grp ^  
  --query 'Parameters[*].[ParameterName,ParameterValue]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  [  
    "audit_logs",  
    "disabled"  
  ],  
  [  
    "tls",  
    "enabled"  
  ],  
  [  
    "ttl_monitor",  
    "enabled"  
  ]  
]
```

修改 Amazon DocumentDB 群組參數組的參數

您可以使用 AWS Management Console 或 AWS CLI 修改參數群組的參數。

Using the AWS Management Console

更新叢集參數群組的參數

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 Parameter groups (參數群組) 窗格中，選擇您想要更新其中參數的叢集參數群組。
4. 結果頁面會顯示這個叢集參數群組的參數，及其對應詳細資訊。選取要更新的參數。
5. 在頁面右上角，選擇 Edit (編輯) 以變更此參數值。如需叢集參數類型的詳細資訊，請參閱 [Amazon DocumentDB 群集參數參考](#)。
6. 進行變更，然後選擇 Modify cluster parameter (修改叢集參數) 以儲存變更。若要捨棄變更，請選擇 Cancel (取消)。

Using the AWS CLI

若要修改叢集參數群組的參數，請使用 `modify-db-cluster-parameter-group` 操作搭配下列參數。

- **--db-cluster-parameter-group-name** – 必要。您想要修改的叢集參數群組名稱。
- **--parameters** – 必要。您正在修改的一或多個參數。每個參數項目必須包括下列：
 - **ParameterName**— 您要修改的參數名稱。
 - **ParameterValue**— 此參數的新值。
 - **ApplyMethod**— 套用此參數變更的方式。允許值為 `immediate` 和 `pending-reboot`。

Note

`static` 必須擁有使用 `ApplyType` 的參數 `ApplyMethod pending-reboot`。

變更叢集參數群組的參數值 (AWS CLI)

下列範例會變更 `tls` 參數。

1. 列出參數和其 **sample-parameter-group** 的值

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name sample-parameter-group
```

針對 Windows：

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "Parameters": [  
    {  
      "Source": "system",  
      "ApplyType": "static",  
      "AllowedValues": "disabled,enabled",  
      "ParameterValue": "enabled",  
      "ApplyMethod": "pending-reboot",  
      "DataType": "string",  
      "ParameterName": "tls",  
      "IsModifiable": true,  
      "Description": "Config to enable/disable TLS"  
    },  
    {  
      "Source": "user",  
      "ApplyType": "dynamic",  
      "AllowedValues": "disabled,enabled",
```

```

        "ParameterValue": "enabled",
        "ApplyMethod": "pending-reboot",
        "DataType": "string",
        "ParameterName": "ttl_monitor",
        "IsModifiable": true,
        "Description": "Enables TTL Monitoring"
    }
]
}

```

2. 修改 **tls** 參數，讓其值為 **disabled**。您無法修改 `ApplyMethod`，因為 `ApplyType` 是 `static`。

若為 Linux、macOS 或 Unix：

```

aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name sample-parameter-group \
  --parameters
  "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"

```

針對 Windows：

```

aws docdb modify-db-cluster-parameter-group ^
  --db-cluster-parameter-group-name sample-parameter-group ^
  --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"

```

此操作的輸出將會如下所示 (JSON 格式)。

```

{
  "DBClusterParameterGroupName": "sample-parameter-group"
}

```

3. 等待至少 5 分鐘。
4. 列出 **sample-parameter-group** 的參數值。

若為 Linux、macOS 或 Unix：

```

aws docdb describe-db-cluster-parameters \
  --db-cluster-parameter-group-name sample-parameter-group

```

針對 Windows :

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "Parameters": [  
    {  
      "ParameterName": "audit_logs",  
      "ParameterValue": "disabled",  
      "Description": "Enables auditing on cluster.",  
      "Source": "system",  
      "ApplyType": "dynamic",  
      "DataType": "string",  
      "AllowedValues": "enabled,disabled",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot"  
    },  
    {  
      "ParameterName": "tls",  
      "ParameterValue": "disabled",  
      "Description": "Config to enable/disable TLS",  
      "Source": "user",  
      "ApplyType": "static",  
      "DataType": "string",  
      "AllowedValues": "disabled,enabled",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot"  
    }  
  ]  
}
```

了解亞馬遜文檔數據庫端點

您可以使用 Amazon 文件資料庫 (與 MongoDB 相容性) 端點連線到叢集或執行個體。亞馬遜 DocumentDB 有三種不同類型的端點，每個端點都有自己的目的。

主題

- [尋找叢集端點](#)
- [搜尋執行個體的端點](#)
- [連接至端點](#)

叢集端點

叢集端點是 Amazon DocumentDB 叢集的端點，可連接至叢集的目前主要執行個體。每個 Amazon DocumentDB 叢集都有一個叢集端點和一個主要執行個體。如果發生容錯移轉，叢集端點會重新對應至新的主要執行個體。

讀取器端點

讀取器端點是 Amazon DocumentDB 叢集的端點，可連接至該叢集的其中一個可用複本。每個 Amazon 文件資料庫叢集都有一個讀取器端點。如果有多個複本，讀取器端點會將每個連線請求導向至其中一個 Amazon DocumentDB 複本。

執行個體端點

執行個體端點會連接至特定的執行個體。叢集中的每個執行個體，不論是主要或複本執行個體，都有自己的唯一執行個體端點。最好不要在您的應用程式中使用執行個體端點。這是因為它們可在容錯移轉時變更角色，因此需要在您的應用程式中進程式碼變更。

尋找叢集端點

您可以使用 Amazon DocumentDB 主控台尋找叢集的叢集端點和讀取器端點，或AWS CLI。

Using the AWS Management Console

使用主控台尋找叢集端點

1. 登入AWS Management Console，然後在以下位置打開亞馬遜文檔數據庫控制台<https://console.aws.amazon.com/docdb>。
2. 在導覽窗格中，選擇 clusters (叢集)。
3. 從叢集清單中選擇您有興趣之叢集的名稱。
4. 向下捲動到 Details (詳細資訊) 區段，並找出叢集端點和讀取器端點。

Cluster details

Configurations and status

ARN
arn:aws:rds:us-east-1: :cluster:docdb-2019-01-09-23-55-38

Cluster identifier
docdb-2019-01-09-23-55-38 (available)

Cluster creation time
1/9/2019, 3:56:03 PM UTC-8

Cluster endpoint
docdb-2019-01-09-23-55-38.cluster- .us-east-1.docdb.amazonaws.com

Reader endpoint
docdb-2019-01-09-23-55-38.cluster-ro- .us-east-1.docdb.amazonaws.com

- 若要連接至這個叢集，捲動到 Connect (連接) 區段。找出 mongo 殼層的連線字串及可用在應用程式碼的連線字串，以連接至您的叢集。

Connect

Connect to this cluster with the mongo shell

```
mongo --ssl --host sample-cluster.cluster-corcjozrlsfc.us-east-1.rds.amazonaws.com:27017 --sslCAFile rds-combined-ca-bundle.pem --username --password <insertYourPassword>
```

Connect to this Chimera cluster with a connection string

```
mongodb:// :<insertYourPassword>@sample-cluster.cluster-corcjozrlsfc.us-east-1.rds.amazonaws.com:27017/?replicaSet=rs0&ssl_ca_certs=rds-combined-ca-bundle.pem
```

Using the AWS CLI

使用 AWS CLI 為您的叢集尋找叢集和讀取器端點時，請執行 `describe-db-clusters` 命令並搭配這些參數。

參數

- **--db-cluster-identifier** 選擇性。指定要傳回其端點的叢集。如果省略，則最多會傳回您的 100 個叢集的端點。
- **--query** 選擇性。指定顯示的欄位。有助於減少您尋找端點時需要檢視的資料量。如果省略，將會傳回叢集的所有資訊。
- **--region** 選擇性。使用 `--region` 參數來指定您要套用命令的區域。若省略，則使用您的預設區域。

Example

以下範例傳回 `sample-cluster` 的 `DBClusterIdentifier`、端點 (叢集端點)，以及 `ReaderEndpoint`。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \  
  --region us-east-1 \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

針對 Windows：

```
aws docdb describe-db-clusters ^  
  --region us-east-1 ^  
  --db-cluster-identifier sample-cluster ^  
  --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  [  
    "sample-cluster",  
    27017,  
    "sample-cluster.cluster-corlsfccjozr.us-east-1.docdb.amazonaws.com",  
    "sample-cluster.cluster-ro-corlsfccjozr.us-east-1.docdb.amazonaws.com"  
  ]  
]
```

現在，有了叢集端點，您可以使用 `mongo` 或 `mongodb` 連接至該叢集。如需詳細資訊，請參閱[連接至端點](#)。

搜尋執行個體的端點

您可以使用 Amazon DocumentDB 主控台或 AWS CLI。

Using the AWS Management Console

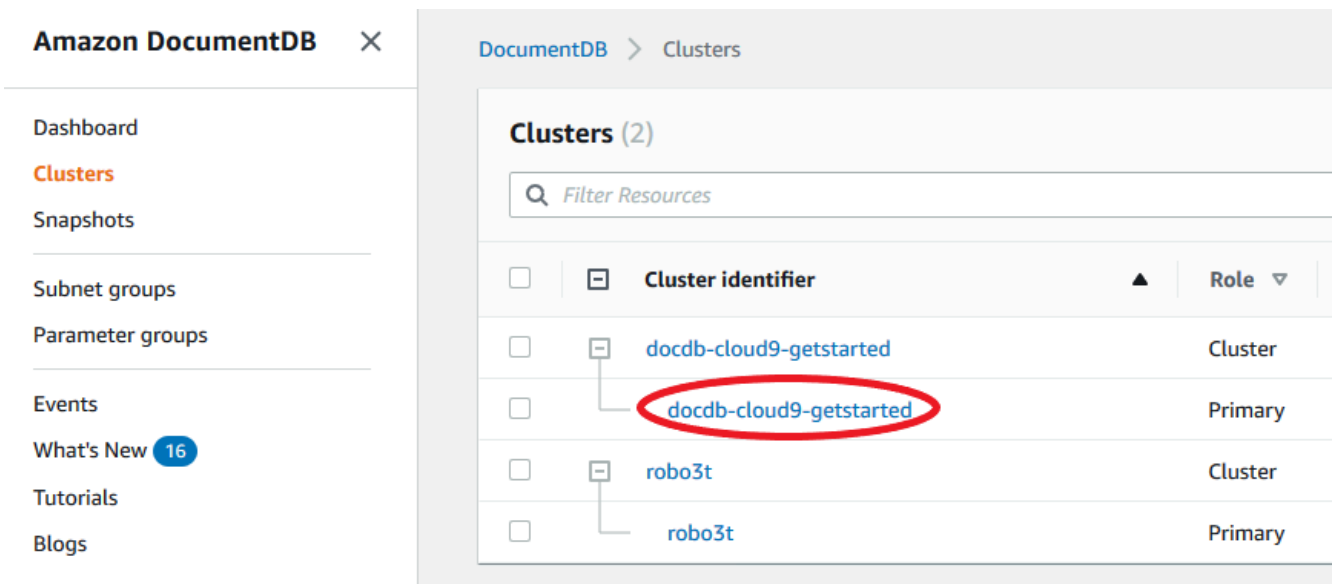
使用主控台尋找執行個體的端點

1. 登入AWS Management Console，然後在以下位置打開亞馬遜文檔數據庫控制台<https://console.aws.amazon.com/docdb>。
2. 在導覽窗格中，選擇 Clusters (叢集)。

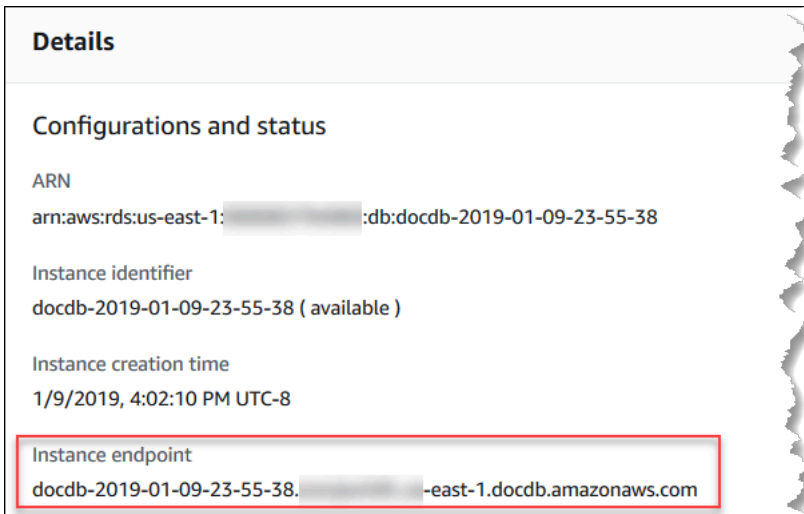
Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 [叢集] 導覽方塊中，您會看到該欄叢集識別碼。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。



4. 勾選您感興趣的執行個體左側的核取方塊。
5. 向下捲動到 Details (詳細資訊) 區段，然後找到執行個體端點。



6. 若要連接至這個執行個體，捲動到 Connect (連接) 區段。找出 mongo 殼層的連線字串及可用在應用程式碼的連線字串，以連接至您的執行個體。



Using the AWS CLI

若要使用 AWS CLI 尋找執行個體端點，請搭配這些引數執行以下命令。

引數

- **--db-instance-identifier**選擇性。指定要傳回其端點的執行個體。如果省略，則最多會傳回您的 100 個執行個體的端點。
- **--query**選擇性。指定顯示的欄位。有助於減少您尋找端點時需要檢視的資料量。如果省略，將會傳回執行個體的所有資訊。Endpoint 欄位有三個成員，若如以下範例將其列入查詢中，則會傳回全部三個成員。如果您只對某些 Endpoint 成員有興趣，請將查詢中的 Endpoint 取代為您有興趣的成員，如第二個範例所示。
- **--region**選擇性。使用 --region 參數來指定您要套用命令的區域。若省略，則使用您的預設區域。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \  
  --region us-east-1 \  
  --db-instance-identifier sample-cluster-instance \  
  --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

針對 Windows :

```
aws docdb describe-db-instances ^  
  --region us-east-1 ^  
  --db-instance-identifier sample-cluster-instance ^  
  --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  [  
    "sample-cluster-instance",  
    {  
      "Port": 27017,  
      "Address": "sample-cluster-instance.corcjozrlsfc.us-  
east-1.docdb.amazonaws.com",  
      "HostedZoneId": "Z2R2ITUGPM61AM"  
    }  
  ]  
]
```

您可以藉由指定 `Endpoint.Port` 和 `Endpoint.Address` 來修改您的查詢，降低輸出以消除端點的 `HostedZoneId`。

若為 Linux、macOS 或 Unix :

```
aws docdb describe-db-instances \  
  --region us-east-1 \  
  --db-instance-identifier sample-cluster-instance \  
  --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

針對 Windows :

```
aws docdb describe-db-instances ^  
  --region us-east-1 ^  
  --db-instance-identifier sample-cluster-instance ^
```

```
--query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[
  [
    "sample-cluster-instance",
    27017,
    "sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com"
  ]
]
```

現在，有了執行個體端點，您可以使用 `mongo` 或 `mongodb` 連接至該執行個體。如需詳細資訊，請參閱[連接至端點](#)。

連接至端點

有了端點之後，您可以使用 `mongo` 殼層或連線字串連接至該叢集或執行個體。

使用 `mongo` 殼層進行連接

使用以下結構來建構您需要的字串，以使用 `mongo` 殼層連接至叢集或執行個體：

```
mongo \  
  --ssl \  
  --host Endpoint:Port \  
  --sslCAFile global-bundle.pem \  
  --username UserName \  
  --password Password
```

`mongo` 殼層範例

連接至叢集：

```
mongo \  
  --ssl \  
  --host sample-cluster.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \  
  --sslCAFile global-bundle.pem \  
  --username UserName \  
  --password Password
```

連接至執行個體：

```
mongo \  
  --ssl \  
  --host sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \  
  --sslCAFile global-bundle.pem \  
  --username UserName \  
  --password Password
```

使用連線字串進行連接

使用以下結構來建構您需要的連線字串，以連接至叢集或執行個體。

```
mongodb://UserName:Password@endpoint:port?replicaSet=rs0&ssl_ca_certs=global-  
bundle.pem
```

連線字串範例

連接至叢集：

```
mongodb://UserName:Password@sample-cluster.cluster-corlsfccjozr.us-  
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

連接至執行個體：

```
mongodb://UserName:Password@sample-cluster-instance.cluster-corlsfccjozr.us-  
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

了解 Amazon DocumentDB Amazon 資源名稱 (ARN)

您在中建立的每個資源 AWS 都會以 Amazon 資源名稱 (ARN) 唯一識別。對於某些 Amazon DocumentDB (與 MongoDB 相容性) 作業，您必須透過指定資源的 ARN 來唯一識別 Amazon DocumentDB 資源。例如，當您新增標籤至資源時，您必須提供資源的 ARN。

主題

- [構建一個 ARN Amazon DocumentDB 資源](#)

- [查找 Amazon DocumentDB 資源 ARN](#)

構建一個 ARN Amazon DocumentDB 資源

您可以使用以下語法為 Amazon DocumentDB 資源構造 ARN。Amazon DocumentDB 共享 Amazon Relational Database Service (Amazon RDS) ARNS 的格式。Amazon DocumentDB ARN 包含和不包含 rds。docdb

arn:aws:rds:*region*:*account_number*:*resource_type*:*resource_id*

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------|----------------|-----------|
| 美國東部 (俄亥俄) | us-east-2 | 3 |
| 美國東部 (維吉尼亞北部) | us-east-1 | 6 |
| 美國西部 (奧勒岡) | us-west-2 | 4 |
| 南美洲 (聖保羅) | sa-east-1 | 3 |
| 亞太區域 (香港) | ap-east-1 | 3 |
| 亞太區域 (海德拉巴) | ap-south-2 | 3 |
| 亞太區域 (孟買) | ap-south-1 | 3 |
| 亞太區域 (首爾) | ap-northeast-2 | 4 |
| 亞太區域 (新加坡) | ap-southeast-1 | 3 |
| 亞太區域 (悉尼) | ap-southeast-2 | 3 |
| 亞太區域 (東京) | ap-northeast-1 | 3 |
| 加拿大 (中部) | ca-central-1 | 3 |
| 中國 (北京) 區域 | cn-north-1 | 3 |
| 中國 (寧夏) | cn-northwest-1 | 3 |

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------------|---------------|-----------|
| 歐洲 (法蘭克福) | eu-central-1 | 3 |
| 歐洲 (愛爾蘭) | eu-west-1 | 3 |
| 歐洲 (倫敦) | eu-west-2 | 3 |
| 歐洲 (米蘭) | eu-south-1 | 3 |
| Europe (Paris) | eu-west-3 | 3 |
| 中東 (阿拉伯聯合大公國) | me-central-1 | 3 |
| AWS GovCloud (美國西部) | us-gov-west-1 | 3 |
| AWS GovCloud (美國東部) | us-gov-east-1 | 3 |

Note

Amazon DocumentDB 架構將儲存和運算分開。對於儲存層，Amazon DocumentDB 會跨三個 AWS 可用區域 (AZ) 複寫六個資料副本。上表中所列的可用區域是您可在指定區域用於佈建運算執行個體的可用區域數量。舉例來說，如果您要在 ap-northeast-1 中啟動 Amazon DocumentDB 叢集，您的儲存將會在三個 AZ 之間進行六種複寫方式，但您的運算執行個體只能在兩個 AZ 中使用。

下表顯示針對特定 Amazon DocumentDB 資源建構 ARN 時應使用的格式。Amazon DocumentDB 共享 Amazon RDS ARN 的格式。Amazon DocumentDB ARN 包含和不包含 rds.docdb

| 資源類型 | ARN 格式/範例 |
|-----------|--|
| 執行個體 (db) | arn:aws:rds: <i>region</i> : <i>account_number</i> :db: <i>resource_id</i> |

| 資源類型 | ARN 格式/範例 |
|--------------------------|--|
| | <pre>arn:aws:rds:us-east-1: 1234567890 :db:sample-db-instance</pre> |
| 叢集 (cluster) | <pre>arn:aws:rds: region:account_number :cluster:resource_id</pre> <pre>arn:aws:rds:us-east-1: 1234567890 :cluster: sample-db-cluster</pre> |
| 叢集參數群組 (cluster-pg) | <pre>arn:aws:rds: region:account_number :cluster-pg:resource_id</pre> <pre>arn:aws:rds:us-east-1: 1234567890 :cluster-pg: sample-db-cluster-parameter-group</pre> |
| 安全群組 (secgrp) | <pre>arn:aws:rds: region:account_number :secgrp:resource_id</pre> <pre>arn:aws:rds:us-east-1: 1234567890 :secgrp:sample-public-secgrp</pre> |
| 叢集快照 (cluster-snapshot) | <pre>arn:aws:rds: region:account_number :cluster-snapshot:resource_id</pre> <pre>arn:aws:rds:us-east-1: 1234567890 :cluster-snapshot: sample-db-cluster-snapshot</pre> |
| 子網路群組 (subgrp) | <pre>arn:aws:rds: region:account_number :subgrp:resource_id</pre> <pre>arn:aws:rds:us-east-1: 1234567890 :subgrp:sample-subnet-10</pre> |

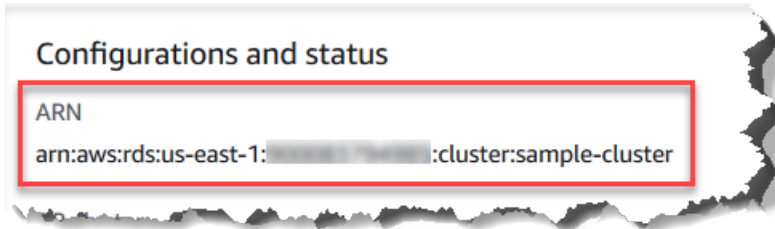
查找 Amazon DocumentDB 資源 ARN

您可以使用或找到 Amazon DocumentDB 資源的 ARN。AWS Management Console AWS CLI

Using the AWS Management Console

若要使用主控台尋找 ARN，請瀏覽至您需要 ARN 的資源，然後檢視該資源的詳細資訊。

例如，您可以在叢集的 Details (詳細資訊) 窗格中取得叢集的 ARN，如下方螢幕擷取畫面所示。



Using the AWS CLI

若要取得使用特定 Amazon DocumentDB 資源的 ARN，請使用該資源的 `describe` 作業。AWS CLI 下表顯示每個 AWS CLI 作業，以及用來取得 ARN 之作業的 ARN 屬性。

| AWS CLI 命令 | ARN 屬性 |
|---|---|
| <code>describe-db-instances</code> | <code>DBInstanceArn</code> |
| <code>describe-db-clusters</code> | <code>DBClusterArn</code> |
| <code>describe-db-parameter-groups</code> | <code>DBParameterGroupArn</code> |
| <code>describe-db-cluster-parameter-groups</code> | <code>DBClusterParameterGroupArn</code> |
| <code>describe-db-security-groups</code> | <code>DBSecurityGroupArn</code> |
| <code>describe-db-snapshots</code> | <code>DBSnapshotArn</code> |
| <code>describe-db-cluster-snapshots</code> | <code>DBClusterSnapshotArn</code> |
| <code>describe-db-subnet-groups</code> | <code>DBSubnetGroupArn</code> |

Example - 尋找叢集的 ARN

下列 AWS CLI 作業會尋找叢集 `sample-cluster` 的 ARN。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].DBClusterArn'
```

針對 Windows：

```
aws docdb describe-db-clusters ^\  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].DBClusterArn'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster"  
]
```

Example - 尋找多個參數群組的 ARN

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameter-groups \  
  --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

針對 Windows：

```
aws docdb describe-db-cluster-parameter-groups ^\  
  --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

此操作的輸出將會如下所示 (JSON 格式)。

```
[  
  "arn:aws:rds:us-east-1:123456789012:cluster-pg:custom3-6-param-grp",  
  "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.aurora5.6",  
  "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.docdb3.6"  
]
```

標記亞馬遜文檔數據庫資源

您可以使用亞馬遜文檔數據庫 (與 MongoDB 兼容性) 標籤將中繼資料添加到您的亞馬遜文檔數據庫資源。這些標籤可以與一起使用AWS Identity and Access Management管理 Amazon DocumentDB 資源存取權限的 (IAM) 政策，以及控制哪些動作可套用至資源。您也可以使用標籤將類似的標記資源分類為群組以追蹤成本。

您可以標記以下亞馬遜文檔數據庫資源：

- 叢集
- 執行個體
- 快照
- 叢集快照
- 參數群組
- 叢集參數群組
- 安全群組
- 子網路群組

亞馬遜文檔數據庫資源標籤概述

亞馬遜文檔數據庫標籤是您定義並與亞馬遜文檔數據庫資源關聯的名稱-值對。此名稱叫做金鑰。為金鑰提供值是選用的。您可以使用標籤將任意資訊指派給 Amazon 文件資源。例如，您可以使用標籤金鑰來定義類別，此標籤值可以是該類別中的項目。例如，您可以定義的標籤鍵project和一個標籤值Salix，表示亞馬遜文檔數據庫資源已分配給 Salix 項目。您也可以使用標籤，透過使用金鑰，將 Amazon DocumentDB 資源指定為用於測試或生產environment=test或者environment=production。我們建議您使用一組一致的標籤金鑰，以便更輕鬆地追蹤與 Amazon DocumentDB 資源相關聯的中繼資料。

您可以使用標籤整理您的 AWS 帳單，藉以反映您自己的成本結構。因此，請進行註冊以取得包含標籤鍵值的 AWS 帳戶帳單。接著，若要查看合併資源的成本，請根據具有相同標籤鍵值的資源來整理您的帳單資訊。例如，您可以使用特定應用程式名稱來標記數個資源，然後整理帳單資訊以查看該應用程式跨數項服務的總成本。如需詳細資訊，請參閱[使用成本配置標記](#)在AWS帳單與成本管理使用者指南。

每個 Amazon DocumentDB 資源都有一個標籤集，其中包含指派給該資源的所有標籤。標籤集最多可以包含 10 個標籤，也可以是空的。如果您將標籤新增至 Amazon DocumentDB 資源，該資源的金鑰與資源上的現有標籤具有相同的金鑰，則新值會覆寫舊值。

AWS 不會將任何語意套用至您的標籤；標籤將嚴格解釋為字元字串。Amazon DocumentDB 可以根據您在建立資源時使用的設定，在執行個體或其他 Amazon DocumentDB 資源上設定標籤。例如，Amazon DocumentDB 可能會新增一個標籤，指出執行個體用於生產或進行測試。

您可以將標籤新增至快照，但是您的帳單將不會反映這個分組。

您可以使用AWS Management Console或AWS CLI在亞馬遜文檔數據庫資源上添加，列出和刪除標籤。使用 AWS CLI 時，您必須提供要處理之資源的 Amazon Resource Name (ARN)。如需 Amazon 文件資料庫 ARN 的詳細資訊，請參閱[了解 Amazon DocumentDB Amazon 資源名稱 \(ARN \)](#)。

標籤限制

下列限制適用於 Amazon 文件資料庫標籤：

- 每個資源的標籤數上限-10
- Key (索引鍵) 長度上限 - 128 個 Unicode 字元
- Value (值) 長度上限 - 256 個 Unicode 字元
- Key (索引鍵) 和 Value (值) 的有效字元 - UTF-8 字元集中的大小寫字母、數字、空格和以下字元：_ . : / = + - 和 @ (Java regex : "`^([\p{L}\p{Z}\p{N}_.:/=+\-]*)$`")
- 標籤鍵與值皆區分大小寫。
- 不要將 aws: 字首用於標籤索引鍵或值；它保留供 AWS 使用。

在亞馬遜文檔數據庫資源上添加和更新標籤

您最多可以將 10 個標籤新增至資源AWS Management Console或AWS CLI。

Using the AWS Management Console

無論您要將標籤新增至何種資源，將標籤新增至資源的程序都很相似。在此範例中，將標籤新增至叢集。

若要使用主控台將標籤新增至叢集或更新標籤

1. 登入AWS Management Console，然後在以下位置打開亞馬遜文檔數據庫控制台<https://console.aws.amazon.com/docdb>。
2. 從導覽窗格選擇 clusters (叢集)。
3. 選擇您要新增標籤至其中的叢集名稱。

4. 向下捲動到 Tags (標籤) 區段，然後選擇 Edit (編輯)。
5. 對於您要新增至此資源的每個標籤，執行以下操作：
 - a. 若要新增新標籤，請在 Key (索引鍵) 方塊中輸入標籤的名稱。若要變更標籤的值，請在 Key (索引鍵) 欄中尋找標籤的名稱。
 - b. 若要給予標籤新值或更新其值，請在 Value (值) 方塊中輸入標籤的值。
 - c. 如果您想新增更多標籤，請選擇 Add (新增)。否則，於完成時選擇 Save (儲存)。

Using the AWS CLI

無論您要將標籤新增至何種資源，將標籤新增至資源的程序都很相似。在此範例中，將三個標籤新增至叢集。第二個標籤 `key2` 沒有值。

使用 AWS CLI 操作 `add-tags-to-resource` 搭配這些參數。

參數

- **--resource-name** 您要新增標籤的亞馬遜文件資料庫資源的 ARN。
- **--tags**— 以格式列出您要新增至此資源的標籤 (鍵值配對) `Key=key-name, Value=tag-value`。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb add-tags-to-resource \  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \  
  --tags Key=key1,Value=value1 Key=key2 Key=key3,Value=value3
```

針對 Windows：

```
aws docdb add-tags-to-resource ^  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \  
  --tags Key=key1,Value=value1 Key=key2 Key=key3,Value=value3
```

`add-tags-to-resource` 操作不會產生輸出。若要查看操作的結果，請使用 `list-tags-for-resource` 操作。

在亞馬遜文檔數據庫資源上列出標籤

您可以使用AWS Management Console或AWS CLI獲取亞馬遜文檔數據庫資源的標籤列表。

Using the AWS Management Console

無論您要將標籤新增至何種資源，在資源上列出標籤的程序都很相似。在此範例中，列出叢集的標籤。

使用主控台列出叢集標籤

1. 打開亞馬遜文檔數據庫控制台<https://console.aws.amazon.com/docdb>。
2. 從導覽窗格選擇 clusters (叢集)。
3. 選擇您要列出其標籤的叢集名稱。
4. 若要查看此資源上的標籤清單，請向下捲動至 Tags (標籤) 區段。

Using the AWS CLI

無論您要列出何種資源的標籤，在資源上列出標籤的程序都很相似。在此範例中，列出叢集上的標籤。

使用 AWS CLI 操作 `list-tags-for-resource` 搭配這些參數。

參數

- **--resource-name** - 必要項目。您要列出標籤的亞馬遜文檔數據庫資源的 ARN。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb list-tags-for-resource \  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

針對 Windows：

```
aws docdb list-tags-for-resource ^  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "TagList": [
    {
      "Key": "key1",
      "Value": "value1"
    },
    {
      "Key": "key2",
      "Value": ""
    },
    {
      "Key": "key3",
      "Value": "value3"
    }
  ]
}
```

從亞馬遜文檔數據庫資源中刪除標籤

您可以使用AWS Management Console或AWS CLI從亞馬遜文檔數據庫資源刪除標籤。

Using the AWS Management Console

無論您要將標籤新增至何種資源，從資源中移除標籤的程序都很相似。在此範例中，從叢集中移除標籤。

使用主控台將標籤從叢集中移除

1. 打開亞馬遜文檔數據庫控制台 <https://console.aws.amazon.com/docdb>。
2. 從導覽窗格選擇 clusters (叢集)。
3. 選擇您要從中移除標籤的叢集名稱。
4. 向下捲動到 Tags (標籤) 區段，然後選擇 Edit (編輯)。
5. 如果您要移除此資源的所有標籤，請選擇 Remove all (全部移除)。否則，針對您要從此資源中移除的每個標籤，執行以下操作：
 - a. 在 Key (索引鍵) 欄中尋找標籤的名稱。
 - b. 在標籤索引鍵的同一列上選擇 Remove (移除)。

- c. 完成時，選擇 Save (儲存)。

Using the AWS CLI

無論您要從何種資源中移除標籤，從資源中移除標籤的程序都很相似。在此範例中，從叢集中移除標籤。

使用 AWS CLI 操作 `remove-tags-from-resource` 搭配這些參數。

- **--resource-name** - 必要項目。您要從中刪除標籤的亞馬遜文檔數據庫資源的 ARN。
- **--tag-keys** - 必要項目。您要從資源中移除的標籤索引鍵清單。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb remove-tags-from-resource \  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \  
  --tag-keys key1 key3
```

針對 Windows：

```
aws docdb remove-tags-from-resource ^  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \  
  --tag-keys key1 key3
```

`removed-tags-from-resource` 操作不會產生輸出。若要查看操作的結果，請使用 `list-tags-for-resource` 操作。

維護 Amazon DocumentDB

Amazon DocumentDB 定期對 Amazon DocumentDB 資源執行維護。維護作業通常大多是更新資料庫引擎 (叢集維護) 或執行個體基礎作業系統 (OS) (執行個體維護)。資料庫引擎更新是必要的修補程式，包括資料庫引擎的安全性修正、錯誤修正和增強功能。作業系統的更新通常包含安全性修正。雖然作業系統修補程式是選用的，但我們建議您在 Amazon DocumentDB 執行個體可用時立即將其套用至 Amazon DocumentDB 執行個體。

資料庫引擎修補程式要求您短時間將 Amazon DocumentDB 叢集離線。一旦可用，這些修補程式就會自動排定在 Amazon DocumentDB 叢集即將到來的排程維護期間套用。

叢集和執行個體維護都有自己個別的維護時段。您選擇不立即套用的叢集和執行個體修改也會在維護期間套用。根據預設，當您建立叢集時，Amazon DocumentDB 會為叢集和每個個別執行個體指派維護時段。您可以在建立叢集或執行個體時選擇維護時段。您也可以隨時修改維護時段，使符合您的業務排程或實務。我們通常會建議您選擇對應用程式影響最低的維護時段 (例如，在晚上或週末)。本指導方針與您體驗的應用程式類型和使用模式高度相關。

主題

- [Amazon DocumentDB 引擎修補程式的通知](#)
- [檢視擱置中的 Amazon DocumentDB 維護動作](#)
- [套用 Amazon DocumentDB 引擎更新](#)
- [使用者啟動更新](#)
- [管理您的 Amazon DocumentDB 維護窗口](#)
- [使用強制作業系統更新](#)

Amazon DocumentDB 引擎修補程式的通知

您將透過 AWS 主控台中 AWS Health Dashboard (AHD) 中的健全狀況事件和電子郵件，收到所需資料庫引擎修補程式的維護通知。當特定 AWS 區域提供 Amazon DocumentDB 引擎維護修補程式時，該區域中所有受影響的 Amazon DocumentDB 使用者帳戶都會收到受修補程式影響的每個 Amazon DocumentDB 版本的 AHD 和電子郵件通知。您可以在 AWS 主控台的 AHD 的 [排程變更] 區段下檢視這些通知。該通知將包含修補程式可用時間、auto 套用排程、受影響叢集清單以及版本說明等詳細資訊。此通知也會透過電子郵件傳送至 AWS 帳戶的 root 使用者電子郵件地址。

| Open and recent issues (0) | Scheduled changes (1) | Other notifications (10) | Event log | | | |
|--|-----------------------|--------------------------|-----------|--------------------------------------|----------|--------------------------|
| Scheduled changes (1) Table Calendar | | | | | | |
| View upcoming events and ongoing events from the past seven days that might affect your AWS infrastructure, such as scheduled maintenance activities. View scheduled changes that occurred more than 7 days ago. | | | | | | |
| <input type="text" value="Add filter"/> | | | | | | |
| Event | Status | Region / Zone | Info | Start time | End time | Affected resources |
| Docdb DB patch upgrade maintenance scheduled | Ongoing | ap-south-1 | | January 2, 2024 at 10:15:46 PM UTC-8 | | 1 entity |

收到此通知後，您可以選擇在排定的自動套用日期之前將這些引擎修補程式自行套用至 Amazon DocumentDB 叢集。或者，您可以等待引擎修補程式在即將到來的維護時段期間自動套用 (預設選項)。

Note

在發行具有新引擎修補程式版本的新 Amazon DocumentDB 引擎修補程式之前，AHD 中的通知狀態將設定為「進行中」。

將引擎修補程式套用至 Amazon DocumentDB 叢集後，叢集的引擎修補程式版本將會更新，以反映通知中的版本。您可以執行 `db.runCommand({getEngineVersion: 1})` 命令來驗證此更新。

AWS Health 還與使用事件建立可擴展的事件驅動應用程式的 Amazon 整合，並與 20 多 EventBridge 個目標整合 AWS Lambda，包括 Amazon Simple Queue Service (SQS) 等。EventBridge 在引擎修補程式可供使用之前，您可以使用 `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_SCHEDULED` 事件代碼來設定 Amazon。當 Amazon DocumentDB 引擎修補程式可用時，您可以設定 EventBridge 回應事件和自動執行動作，例如擷取事件資訊、啟動其他事件 AWS Console Mobile Application、透過其他管道 (例如推送通知) 傳送通知，以及採取更正或其他動作。

在 Amazon DocumentDB 取消引擎修補程式的罕見情況下，您會收到 AHD 通知以及一封電子郵件，通知您有關取消的資訊。因此，您可以使用 `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_CANCELLED` 事件代碼設置 Amazon EventBridge 以響應此事件。查看 Amazon 使用 EventBridge 者指南以進一步了解如何使用 [Amazon EventBridge 規則](#)。

檢視擱置中的 Amazon DocumentDB 維護動作

您可以使用或來檢視叢集是否有維護更新 AWS Management Console 可用 AWS CLI。

如有更新可用，您可執行下列操作之一：

- 延遲目前排定在下一個維護時段執行的維護動作 (僅適用於 OS 修補程式)。
- 立即套用維護動作。
- 排程在下次的維護時段啟動維護動作。

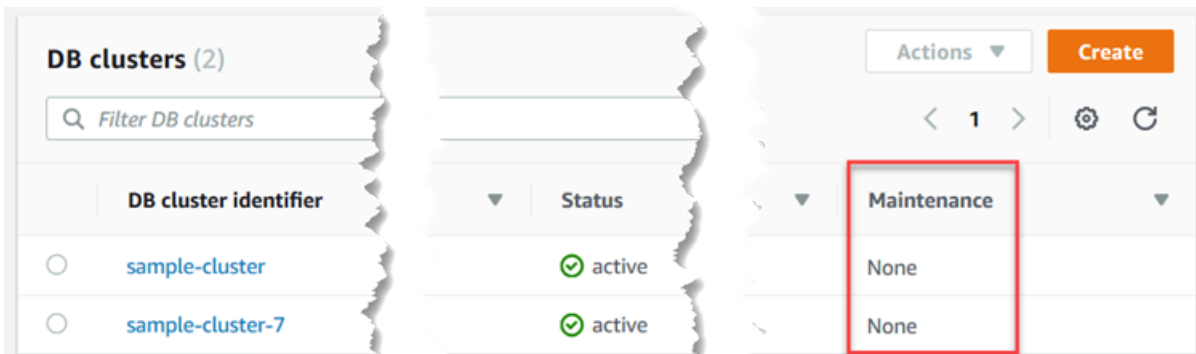
Note

如果您不採取任何動作，必要的維護動作 (例如引擎修補程式) 將會 auto 動套用到即將到來的排程維護時段中。

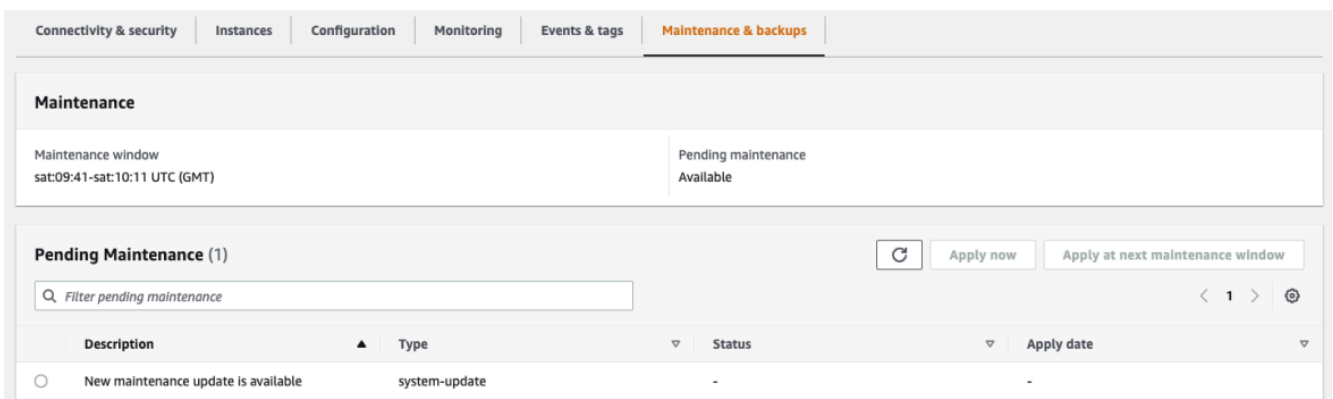
維護時段會判斷待定的操作何時開始，但不限制這些操作的總執行時間。

Using the AWS Management Console

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。
3. 如果有可用的更新，Amazon DocumentDB 主控台上叢集的「維護」欄中會以「可用」、「必要」或「下一個時段」一詞表示，如下所示：



4. 若要採取動作，請選擇叢集以顯示其詳細資料，然後選擇 [維護與備份]。[擱置維護] 項目隨即顯示。



Using the AWS CLI

請使用下列 AWS CLI 作業來決定要擱置的維護動作。此處的輸出顯示無待定的維護動作。

```
aws docdb describe-pending-maintenance-actions
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
```

```
"PendingMaintenanceActions": []  
}
```

套用 Amazon DocumentDB 引擎更新

使用 Amazon DocumentDB，您可以選擇套用維護作業的時機。您可以決定 Amazon DocumentDB 何時使用或套用更新。AWS Management Console AWS CLI

使用本主題中的程序可立即升級或排程叢集的升級。

Using the AWS Management Console

您可以使用主控台來管理 Amazon DocumentDB 叢集的更新。

若要管理叢集的更新

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。
3. 在叢集清單中，選擇您要套用維護操作之叢集名稱旁的按鈕。
4. 在 Actions (動作) 選單上，選擇下列其中一項：
 - Upgrade now (立即升級)，立即執行待處理的維護任務。
 - Upgrade at next window (在下一個時段升級)，在叢集的下一個維護時段執行待處理的維護任務。

或者，您也可以可以在叢集維護與備份索引標籤的擱置維護區段中，按一下 [立即套用] 或 [在下一個維護時段套用] (請參閱上 AWS Management Console 一節使用)。

Note

如果沒有擱置中的維護工作，則上述所有選項都處於非作用中狀態。

Using the AWS CLI

若要將擱置的更新套用至叢集，請使用此 `apply-pending-maintenance-action` AWS CLI 作業。

參數

- **--resource-identifier** 擱置維護動作適用於之資源的 Amazon DocumentDB Amazon 資源名稱 (ARN)。
- **--apply-action** 要套用至此資源的擱置維護動作。

有效值：system-update 和 db-upgrade。

- **--opt-in-type** 指定選擇加入要求類型的值，或復原選擇加入要求。無法復原 immediate 類型的選擇使用請求。

有效值：

- immediate 立即套用維護動作。
- next-maintenance 在資源的下一個維護時段期間套用維護動作。
- undo-opt-in— 取消任何現有的 next-maintenance 選擇加入請求。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb apply-pending-maintenance-action \  
  --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb \  
  --apply-action system-update \  
  --opt-in-type immediate
```

針對 Windows：

```
aws docdb apply-pending-maintenance-action ^  
  --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb ^  
  --apply-action system-update ^  
  --opt-in-type immediate
```

若要傳回至少有一個擱置更新的資源清單，請使用此 describe-pending-maintenance-actions AWS CLI 作業。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb describe-pending-maintenance-actions \  
  --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb
```

```
--resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

針對 Windows :

```
aws docdb describe-pending-maintenance-actions ^  
--resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "PendingMaintenanceActions": [  
    {  
      "ResourceIdentifier": "arn:aws:rds:us-  
east-1:001234567890:cluster:sample-cluster",  
      "PendingMaintenanceActionDetails": [  
        {  
          "Action": "system-update",  
          "CurrentApplyDate": "2019-01-11T03:01:00Z",  
          "Description": "db-version-upgrade",  
          "ForcedApplyDate": "2019-01-18T03:01:00Z",  
          "AutoAppliedAfterDate": "2019-01-11T03:01:00Z"  
        }  
      ]  
    }  
  ]  
}
```

您也可以透過指定作業的 `--filters` 參數，傳回叢集的資源清單 `describe-pending-maintenance-actions` AWS CLI 單。 `--filters` 操作的格式為 `Name=filter-name, Values=resource-id, ...`。

`db-cluster-id` 是篩選器 `Name` 參數的可接受值。此值接受叢集識別碼或 ARN 的清單。在系統所傳回的叢集待處理維護動作清單中，只會包含以這些識別碼或 ARN 識別的項目。

下方範例將傳回 `sample-cluster1` 與 `sample-cluster2` 叢集的待處理維護動作。

Example

若為 Linux、macOS 或 Unix :

```
aws docdb describe-pending-maintenance-actions \  

```

```
--filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

針對 Windows :

```
aws docdb describe-pending-maintenance-actions ^  
--filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

申請日期

每個維護動作都有各自的套用日期，您可以在描述待定維護動作時找到。當您從讀取擱置維護動作的輸出時 AWS CLI，會列出三個日期：

- **CurrentApplyDate**立即套用維護動作或在下一個維護時段套用的日期。如果維護是選用的，則此值可以是 null。
- **ForcedApplyDate**自動套用維護的日期，與您的維護時段無關。
- **AutoAppliedAfterDate**在叢集維護時段期間套用維護的日期之後。

使用者啟動更新

身為 Amazon DocumentDB 使用者，您可以啟動叢集或執行個體的更新。例如，您可以將執行個體的類別修改為具有或多或更少記憶體類別，也可以變更叢集的參數群組。Amazon DocumentDB 檢視這些變更與 Amazon DocumentDB 啟動的更新不同。如需修改叢集或執行個體的詳細資訊，請參閱下列內容：

- [修改 Amazon DocumentDB 叢集](#)
- [修改 Amazon DocumentDB 實例](#)

若要查看待定的使用者啟動修改內容清單，請執行下列命令。

Example

查看執行個體的待定使用者啟動變更

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \
```



```
--query 'DBInstances[*].  
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

針對 Windows :

```
aws docdb describe-db-instances ^  
  --query 'DBInstances[*].  
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

此操作的輸出將會如下所示 (JSON 格式)。

在本案例中，sample-cluster-instance 有 db.r5.xlarge 執行個體類別的待定變更，而 sample-cluster-instance-2 沒有待定變更。

```
[  
  [  
    "sample-cluster",  
    "sample-cluster-instance",  
    {  
      "DBInstanceClass": "db.r5.xlarge"  
    }  
  ],  
  [  
    "sample-cluster",  
    "sample-cluster-instance-2",  
    {}  
  ]  
]
```

管理您的 Amazon DocumentDB 維護窗口

每個執行個體和叢集的維護時段為每週一次，在此期間會套用任何待定的變更。維護時段可以在請求或要求事件中，控制修改及軟體修補作業的時機。若在特定某週排定維護事件，系統將在您指定的 30 分鐘維護時段內啟動該事件。此外，多數維護事件也能在 30 分鐘的維護時段內完成，但較大型的維護事件可能需要 30 分鐘以上才能完成。

30 分鐘的維護時段是從每個區域之 8 小時時段內隨機選取的。如果您在建立執行個體或叢集時未指定偏好的維護時段，Amazon DocumentDB 會在一週中隨機選取的日期指派 30 分鐘的維護時段。

下表列出每個區域的時段，預設維護時段會從此時段中指派。

| 區域名稱 | 區域 | UTC 時間區塊 |
|---------------|----------------|-----------------|
| 美國東部 (俄亥俄) | us-east-2 | 03:00-11:00 |
| 美國東部 (維吉尼亞北部) | us-east-1 | 03:00-11:00 |
| 美國西部 (奧勒岡) | us-west-2 | 06:00-14:00 |
| 亞太區域 (香港) | ap-east-1 | 06:00-14:00 |
| 亞太區域 (海德拉巴) | ap-south-2 | 上午 6 時半至下午 2 時半 |
| 亞太區域 (孟買) | ap-south-1 | 06:00-14:00 |
| 亞太區域 (首爾) | ap-northeast-2 | 13:00-21:00 |
| 亞太區域 (新加坡) | ap-southeast-1 | 下午 2 時 |
| 亞太區域 (悉尼) | ap-southeast-2 | 下午八時 |
| 亞太區域 (東京) | ap-northeast-1 | 13:00-21:00 |
| 加拿大 (中部) | ca-central-1 | 03:00-11:00 |
| 中國 (北京) | cn-north-1 | 06:00-14:00 |
| 中國 (寧夏) | cn-northwest-1 | 06:00-14:00 |
| 歐洲 (法蘭克福) | eu-central-1 | 下午 9 時至 5 時 |
| 歐洲 (愛爾蘭) | eu-west-1 | 22:00-06:00 |
| 歐洲 (倫敦) | eu-west-2 | 22:00-06:00 |
| 歐洲 (米蘭) | eu-south-1 | 02:00-10 |
| 歐洲 (巴黎) | eu-west-3 | 23:59-07:29 |
| 中東 (阿拉伯聯合大公國) | me-central-1 | 下午 5 時至下午 1 時 |
| 南美洲 (聖保羅) | sa-east-1 | 凌晨八時 |

| 區域名稱 | 區域 | UTC 時間區塊 |
|---------------------|---------------|-------------|
| AWS GovCloud (美國東部) | us-gov-east-1 | 下午一時 |
| AWS GovCloud (美國西部) | us-gov-west-1 | 06:00-14:00 |

更改您的 Amazon DocumentDB 維護窗口

維護時段應落在使用量最低的時段，因此可能需要不時進行變更。除非系統套用變更 (例如擴展儲存操作或執行個體類別變更) 及要求停機，否則這段時間無法使用您的叢集或執行個體。而且無法使用的時間，僅為執行必要變更所需的最短時間。

若要升級至資料庫引擎，Amazon DocumentDB 會使用叢集偏好的維護時段，而非個別執行個體的維護時段。

變更維護時段

- 針對叢集：請參閱[修改 Amazon DocumentDB 叢集](#)。
- 針對執行個體：請參閱[修改 Amazon DocumentDB 實例](#)。

使用強制作業系統更新

Amazon DocumentDB 叢集中的執行個體偶爾需要作業系統更新。Amazon DocumentDB 會將作業系統升級至較新版本，以改善資料庫效能和客戶整體安全狀態。作業系統更新不會變更 Amazon DocumentDB 執行個體的叢集引擎版本或執行個體類別。

建議您先更新叢集中的讀取器執行個體，然後再更新寫入器執行個體，以最大化叢集的可用性。我們不建議同時更新讀取器和寫入器執行個體，因為如果發生容錯移轉，可能會產生較長的停機時間。

作業系統更新沒有套用日期，而且可以隨時套用。我們建議您定期套用這些資料庫，以使 Amazon DocumentDB 資料庫保持在最新狀態。Amazon DocumentDB 不會自動應用這些更新。若要在新的選用更新可用時收到通知，您可以訂閱安全修補事件類別中的 RDS-EVENT-0230。如需訂閱 Amazon 文件資料庫事件的相關資訊，請參閱[訂閱 Amazon DocumentDB 事件訂閱](#)。

在叢集或執行個體上執行維護時，如果執行個體為主執行個體，可預期將會容錯移轉。為了提高可用性，我們建議您為 Amazon DocumentDB 叢集使用一個以上的執行個體。如需詳細資訊，請參閱[Amazon DocumentDB 容錯移轉](#)。

Note

對於某些管理功能，Amazon DocumentDB 使用與 Amazon Relational Database Service (Amazon RDS) 共享的操作技術。

Important

在作業系統升級期間，您的 Amazon DocumentDB 執行個體將會離線。

Note

為了履行各種合規義務，可能需要將所有選用與強制更新保持為最新的狀態。我們建議您在維護期間定期套用 Amazon DocumentDB 所提供的所有更新。

您可以使用 AWS Management Console 或 AWS CLI 來判斷更新是選擇性還是必要更新。

Using the AWS Management Console

若要判斷更新是選擇性還是必要的，請使用下列步 AWS Management Console 驟：

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在瀏覽窗格中，選擇 [叢集]，然後選取執行個體。
3. 選擇「維護」。
4. 在「擱置維護」段落中，尋找作業系統更新，然後檢查「狀態」值。

在中 AWS Management Console，作業系統更新的維護狀態設定為可用，而且沒有 [套用] 日期，如下圖所示：

The screenshot shows the AWS Management Console interface for DocumentDB. At the top, there are navigation tabs: Connectivity & security, Configuration, Monitoring, Maintenance (highlighted), and Events & tags. Below the tabs, the 'Maintenance' section displays the current maintenance window as 'tue:07:45-tue:08:15 UTC (GMT)' and indicates that 'Pending maintenance' is 'Available'. A 'Pending Maintenance (1)' section follows, featuring a refresh button, 'Apply now', and 'Apply at' buttons. A search bar labeled 'Filter pending maintenance' is present. Below the search bar is a table with columns for Description, Type, and Status. One entry is visible: 'New Operating System update is available' with a type of 'system-update' and a status of '-'. There are also 'Apply now' and 'Apply at' buttons for this entry.

您可以選取作業系統更新，然後按一下「擱置維護」段落中的「立即套用」或「下一個維護時段套用」。如果維護值是下一個時段，請選擇「延期升級」來延遲維護項目。如果維護動作已經啟動，則無法延遲維護動作。

或者，您也可以按一下導覽窗格中的 [叢集]，從叢集清單中選擇執行個體，然後從 [動作] 功能表中選取 [立即套用] 或 [在下一個維護時套用] 視窗。

Using the AWS CLI

若要使用判斷更新是選用的還是強制更新 AWS CLI，請呼叫以下 `describe-pending-maintenance-actions` 指令：

```
aws docdb describe-pending-maintenance-actions
```

強制作業系統更新包括 `AutoAppliedAfterDate` 值與 `CurrentApplyDate` 值。選用作業系統更新未包含這些值。

下列輸出顯示必要的作業系統更新：

```
{
```

```
"ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb1",
"PendingMaintenanceActionDetails": [
  {
    "Action": "system-update",
    "AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
    "CurrentApplyDate": "2022-08-31T00:00:00+00:00",
    "Description": "New Operating System update is available"
  }
]
}
}
The following output shows an optional operating system update.
{
  "ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

作業系統更新的可用性

作業系統更新僅適用於 Amazon DocumentDB 引擎版本和執行個體類別。因此，Amazon DocumentDB 執行個體會在不同的時間接收或需要更新。當您的執行個體根據其引擎版本和執行個體類別提供作業系統更新時，更新會顯示在主控台中。它也可以通過運行 AWS CLI `describe-pending-maintenance-actions` 命令或調用 `DescribePendingMaintenanceActions` API 操作進行查看。如果您的執行個體有可用的更新，您可以按照[套用 Amazon DocumentDB 更新中的指示](#)來更新作業系統。

了解服務連結角色

亞馬遜文檔數據庫 (與 MongoDB 相容性) 使用 AWS Identity and Access Management (IAM) 服務連結的角色。[服務連結角色](#)是一種直接連結至 Amazon DocumentDB 的唯一 IAM 角色類型。服務連結角色由 Amazon DocumentDB 預先定義，並包含服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓使用 Amazon DocumentDB 變得更容易，因為您不必手動新增必要的許可。Amazon DocumentDB 會定義其服務連結角色的許可，除非另有定義，否則只有 Amazon DocumentDB 可以擔任其角色。定義的許可包括信任政策和許可政策，並且該許可政策不能附加到任何其他 IAM 實體。

您必須先刪除角色的相關資源，才能刪除角色。這樣可以保護您的 Amazon DocumentDB 資源，因為您無法不小心移除存取資源的權限。

如需關於支援服務連結角色的其他服務的資訊，請參閱[可搭配 IAM 運作的 AWS 服務](#)，並尋找 Service-Linked Role (服務連結角色) 欄顯示為 Yes (是) 的服務。選擇具有連結的 Yes (是)，以檢視該服務的服務連結角色文件。

亞馬遜文檔數據庫服務鏈接角色許可

Amazon 文件資料庫 (與 MongoDB 相容性) 使用名為的服務連結角色，AWSServiceRoleForRDS 允許 Amazon DocumentDB 代表叢集呼叫 AWS 服務。

AWSServiceRoleForRDS 服務連結角色信任下列服務以擔任角色：

- `docdb.amazonaws.com`

角色許可政策允許 Amazon DocumentDB 在指定的資源上完成下列動作：

- 在 ec2 上的動作：
 - `AssignPrivateIpAddresses`
 - `AuthorizeSecurityGroupIngress`
 - `CreateNetworkInterface`
 - `CreateSecurityGroup`
 - `DeleteNetworkInterface`
 - `DeleteSecurityGroup`
 - `DescribeAvailabilityZones`
 - `DescribeInternetGateways`
 - `DescribeSecurityGroups`
 - `DescribeSubnets`
 - `DescribeVpcAttribute`
 - `DescribeVpcs`
 - `ModifyNetworkInterfaceAttribute`
 - `RevokeSecurityGroupIngress`
 - `UnassignPrivateIpAddresses`
- 在 sns 上的動作：

- ListTopic
- Publish
- 在 cloudwatch 上的動作：
 - PutMetricData
 - GetMetricData
 - CreateLogStream
 - PullLogEvents
 - DescribeLogStreams
 - CreateLogGroup

Note

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。您可能會遇到下列錯誤訊息：

無法建立資源。請確認您擁有建立服務連結角色的許可。否則請等待，然後再試一次。

如果您看到此錯誤，請確認您已啟用以下許可：

```
{
  "Action": "iam:CreateServiceLinkedRole",
  "Effect": "Allow",
  "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": "rds.amazonaws.com"
    }
  }
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的[服務連結角色許可](#)。

建立亞馬遜文件資料庫服務連結角色

您不需要手動建立一個服務連結角色。當您建立叢集時，Amazon DocumentDB 會為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您建立叢集時，Amazon DocumentDB 會再次為您建立服務連結角色。

修改亞馬遜文件資料庫服務連結角色

亞馬遜文檔數據庫不允許您修改AWSServiceRoleForRDS服務鏈接的角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。不過，您可以使用 IAM 修改角色的說明。如需更多資訊，請參閱 IAM 使用者指南中的[編輯服務連結角色](#)。

刪除 Amazon 文件資料庫服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。不過您必須先刪除您的所有叢集，才能刪除服務連結角色。

清理亞馬遜文檔數據庫服務鏈接角色

您必須先確認服務連結角色沒有作用中的工作階段，並移除該角色使用的資源，之後才能使用 IAM 將其刪除。

檢查服務連結角色是否有使用主控台的作用中工作階段

1. 登入AWS Management Console並開啟 IAM 主控台，位於<https://console.aws.amazon.com/iam/>。
2. 在 IAM 主控台的導覽窗格中，選擇 [角色]，然後選擇AWSServiceRoleForRDS角色的名稱 (而非核取方塊)。
3. 在所選角色的 Summary (摘要) 頁面中，選擇 Access Advisor (存取 Advisor) 分頁。
4. 在 Access Advisor (存取 Advisor) 標籤中，檢閱服務連結角色的近期活動。

Note

如果您不確定 Amazon DocumentDB 是否正在使用該AWSServiceRoleForRDS角色，可以嘗試刪除該角色。如果服務正在使用該角色，則刪除會失敗，而您可以檢視正在使用該角色的區域。如果服務正在使用該角色，您必須先等到工作階段結束，才能刪除該角色。您無法撤銷服務連結角色的工作階段。

如果您想要移除 AWSServiceRoleForRDS 角色，則必須先刪除您所有的執行個體和叢集。如需有關刪除執行個體和叢集的詳細資訊，請參閱下列主題：

- [刪除 Amazon DocumentDB 實例](#)
- [刪除 Amazon DocumentDB 集群](#)

支援 Amazon 文件資料庫服務連結角色的區域

Amazon DocumentDB 支援在所有提供服務的區域中使用服務連結角色。如需詳細資訊，請參閱 <https://docs.aws.amazon.com/documentdb/latest/developerguide/regions-and-azs.html#regions-and-azs-availability>。

使用 Amazon DocumentDB 彈性集群

Amazon DocumentDB 彈性叢集支援工作負載，每秒可達數百萬次讀取/寫入，以及數 PB 的儲存容量。彈性叢集也不需要選擇、管理或升級執行個體，簡化開發人員與 Amazon DocumentDB 互動的方式。

Amazon DocumentDB 彈性叢集的建立目的是：

- 為尋求資料庫的客戶提供解決方案，該資料庫具有豐富的查詢功能和 MongoDB API 相容性，提供幾乎無限規模的資料庫。
- 為客戶提供更高的連線限制，並減少修補所造成的停機時間。
- 持續投資雲端原生、彈性且領先業界的 JSON 工作負載架構。

主題

- [彈性叢集使用案例](#)
- [彈性叢集的優點](#)
- [彈性叢集區域和版本可用性](#)
- [限制](#)
- [Amazon DocumentDB 彈性集群：它是如何工作的](#)
- [開始使用 Amazon DocumentDB 彈性叢集](#)
- [最佳實務](#)
- [管理彈性叢集](#)
- [Amazon DocumentDB Elastic Clusters 的靜態靜態資料加密](#)
- [彈性叢集中的服務連結角色](#)

彈性叢集使用案例

文件資料庫對於需要靈活結構描述以便快速進行重複性開發的工作負載來說非常適合。例如，Amazon DocumentDB 使用案例，請參閱。[文件資料庫使用案例](#)

以下是彈性叢集可提供顯著優勢的一些使用案例範例：

使用者設定檔

由於文件資料庫具有彈性的結構描述，因此它們可以大規模儲存具有不同屬性和資料值的文件。彈性叢集是不同使用者提供不同類型資訊的線上設定檔的實用解決方案。假設您的應用程式支援數億個使用者設定檔。您可以使用彈性叢集來支援此類應用程式，因為這些應用程式可以擴展和擴展，以支援數百萬次寫入和讀取這些使用者設定檔。您也可以縮減離峰時段以降低成本。

內容管理和歷史記錄

為了有效管理內容，您必須能夠收集和彙總各種不同來源的內容，然後將內容傳遞給客戶。文件資料庫採用靈活的結構描述，是收集和存放任何類型資料最理想的選擇。您可以使用它們來創建和合併新類型的內容，包括用戶生成的內容，例如圖像，評論和視頻。隨著時間的推移，您的數據庫可能需要更多存儲。使用彈性叢集，您可以將資料分配到更多儲存磁碟區，以便在單一叢集中儲存數 PB 的資料。

彈性叢集的優點

AWS 服務整合

Amazon DocumentDB 彈性叢集與其他 AWS 服務整合的方式與 Amazon DocumentDB 的方式相同：

- 移轉-您可以使用 AWS Database Migration Service (DMS) 從 MongoDB 和其他關聯式資料庫遷移到 Amazon DocumentDB 彈性叢集。
- 監控-您可以使用 Amazon 監控彈性叢集的運作狀態和效能 CloudWatch。
- 安全性-您可以透過 AWS Identity and Access Management (IAM) 設定身份驗證和授權來管理彈性叢集，並使用 Amazon VPC 進行僅限 VPC 的安全連線。
- 數據管理-您可以使 AWS Glue 用將數據導入和導出到其他 AWS 服務，如 Amazon S3，Amazon Redshift 和 Amazon OpenSearch 服務。

彈性叢集區域和版本可用性

區域可用性

下表顯示目前提供 Amazon DocumentDB 彈性叢集的 AWS 區域，以及每個區域的端點。

| 區域名稱 | 區域 | 可用性區域 |
|---------------|----------------|-------|
| 美國東部 (維吉尼亞北部) | us-east-1 | 5 |
| 美國東部 (俄亥俄) | us-east-2 | 3 |
| 美國西部 (奧勒岡) | us-west-2 | 3 |
| 亞太區域 (孟買) | ap-south-1 | 3 |
| 亞太區域 (首爾) | ap-northeast-2 | 3 |
| 亞太區域 (新加坡) | ap-southeast-1 | 3 |
| 亞太區域 (雪梨) | ap-southeast-2 | 3 |
| 亞太區域 (東京) | ap-northeast-1 | 3 |
| 南美洲 (聖保羅) | sa-east-1 | 3 |
| 歐洲 (法蘭克福) | eu-central-1 | 3 |
| 歐洲 (愛爾蘭) | eu-west-1 | 3 |
| 歐洲 (倫敦) | eu-west-2 | 3 |

版本可用性

彈性叢集支援 MongoDB 5.0 相容的有線通訊協定。如需 DocumentDB 4.0 執行個體型叢集與彈性叢集之間的差異，請參閱 [Amazon DocumentDB 4.0 和彈性叢集之間的功能差異](#)

限制

彈性叢集管理

此版本不支援下列叢集管理功能：

- 能夠建立全域叢集
- 現有的 Amazon DocumentDB 事件和訂閱事件

- 範圍分片
- 碎片現有集合
- 多欄位碎片金鑰
- 更改碎片鍵
- Point-in-time 還原
- 複製
- Performance Insights

Note

如需有關彈性叢集限制的資訊，請參閱[Amazon DocumentDB 配額和限制](#)。

查詢和寫入操作

此發行版本不支援下列查詢和寫入作業指令和功能：

- 縮放作業期間的 DDL 命令
- Profiler
- 參數群組
- AWS Config
- AWS Backup

集合和索引管理

此發行版本不支援下列集合和索引管理功能：

- 稀鬆索引
- 空間索引
- 建立背景索引

管理和診斷

此版本不支援下列管理和診斷命令和功能：

- AWS Secrets Manager
- Role-based-access-control (RBAC) 自訂角色。
- 連線時，不支援寫入 0 的問題。
- 變更屬於目前未指派給現有彈性叢集之 VPC 的子網路。

選擇加入功能

此版本不支援下列 Amazon DocumentDB 選擇加入功能：

- ACID 交易
- DDL/DML 稽核
- Change streams
- 工作階段指

Amazon DocumentDB 彈性集群：它是如何工作的

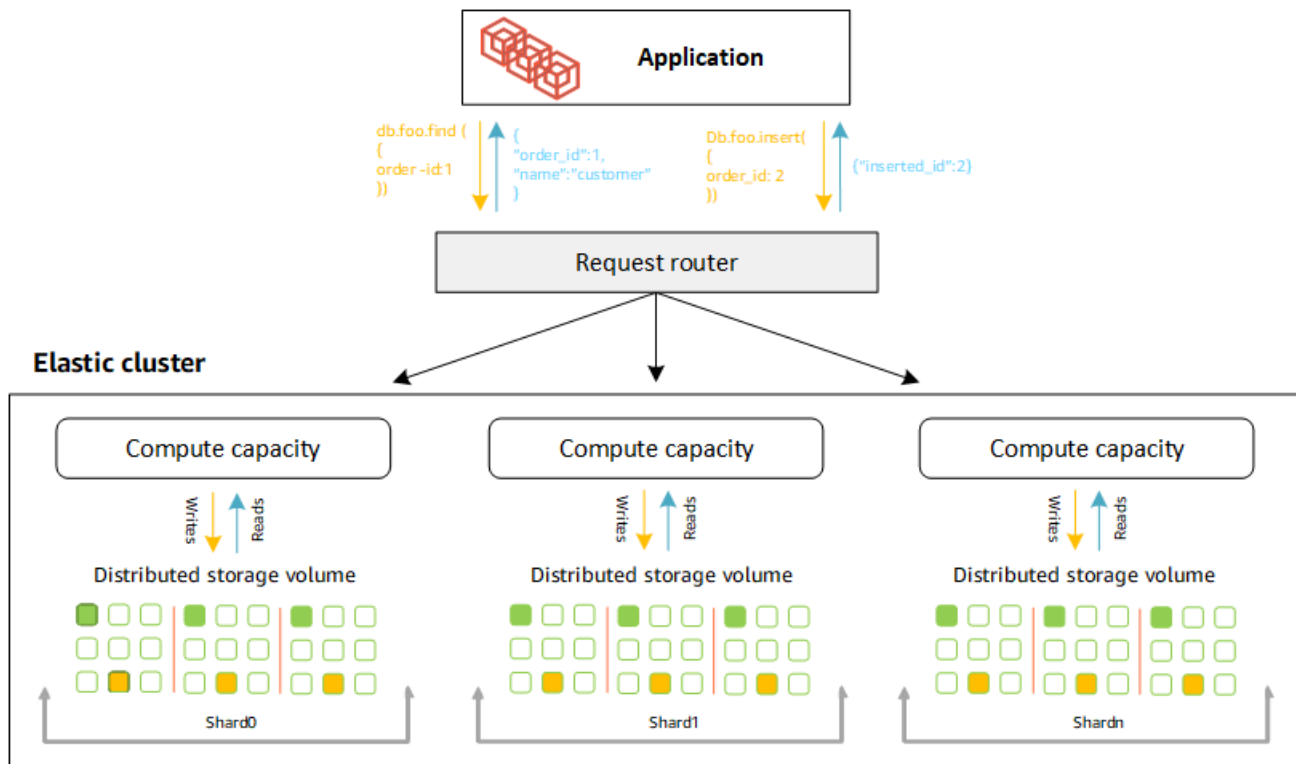
本節中的主題提供支援 Amazon DocumentDB 彈性叢集的機制和功能的相關資訊。

主題

- [Amazon DocumentDB 彈性集群分片](#)
- [彈性叢集移轉](#)
- [彈性叢集擴展](#)
- [彈性群集可靠性](#)
- [彈性叢集儲存與可用性](#)
- [Amazon DocumentDB 4.0 和彈性集群之間的功能差異](#)

Amazon DocumentDB 彈性集群分片

Amazon DocumentDB 彈性叢集使用雜湊式分割，跨分散式儲存系統分割資料。分片 (又稱為分割) 會將大型資料集分割成跨多個節點的小型資料集，讓您能夠在垂直擴展限制之外擴充資料庫。彈性叢集在 Amazon DocumentDB 中使用運算和儲存的分離 (或「解耦」)，讓您能夠彼此獨立擴展。彈性叢集不會透過在計算節點之間移動小塊資料來重新分割集合，而是在分散式儲存系統中有效地複製資料。



碎片定義

碎片命名法的定義：

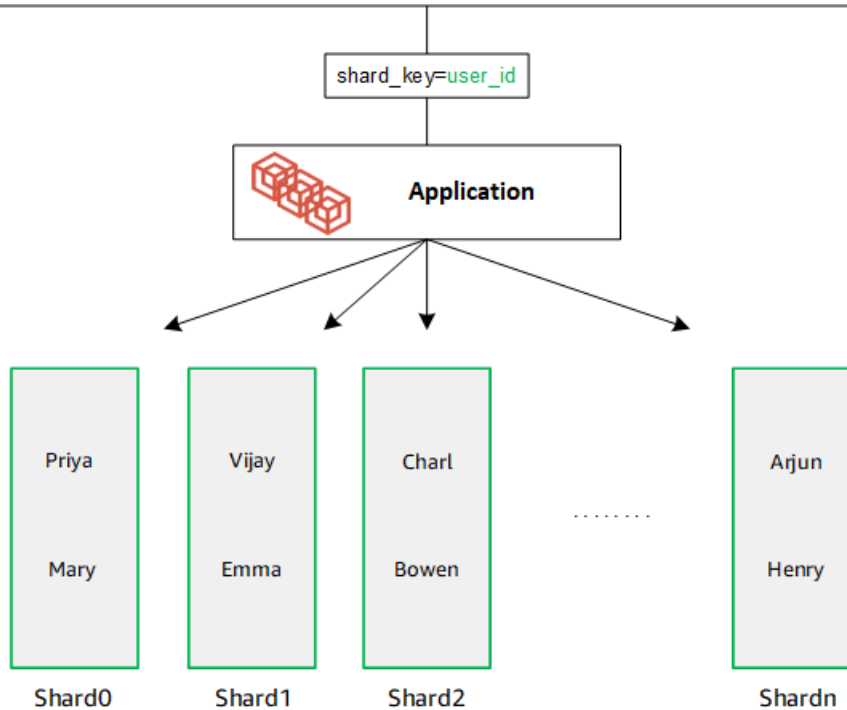
- 碎片-碎片為彈性集群提供計算。分片默認情況下將有兩個節點。您最多可以設定 32 個碎片，且每個碎片最多可以有 64 個 vCPUs。
- 碎片索引鍵 — 碎片索引鍵是分割集合中 JSON 文件中的必填欄位，彈性叢集用來將讀取和寫入流量分配給相符的碎片。
- 碎片集合-碎片集合是一個集合，其數據分佈在數據分區中的彈性集群中。
- 分區 — 分區是碎片數據的邏輯部分。當您建立分割集合時，資料會根據碎片索引鍵自動組織成每個碎片內的分區。每個碎片都有多個分區。

在配置的碎片之間分配數據

創建一個具有許多唯一值的分片鍵。良好的碎片金鑰會在基礎碎片之間平均分割資料，為您的工作負載提供最佳的輸送量和效能。下列範例是使用名為「user_id」的碎片索引鍵的員工姓名資料：

Employee Dataset

```
{ "name": "Priya", "lastname": "Kumar", "role": "Manager", "user_id": 1, "phone": "2223333" }
{ "name": "Mary", "lastname": "Johnson", "role": "Manager", "user_id": 2, "phone": "3334444" }
{ "name": "Vijay", "lastname": "Agarwal", "role": "Manager", "user_id": 3, "phone": "4445555" }
{ "name": "Emma", "lastname": "Wu", "role": "SW Architect", "user_id": 4, "phone": "6667777" }
{ "name": "Charl", "lastname": "Van rooyen", "role": "SW Architect", "user_id": 5, "phone": "7778888" }
{ "name": "Bowen", "lastname": "Chen", "role": "SW Developer", "user_id": 6, "phone": "8889999" }
{ "name": "Arjun", "lastname": "Reddy", "role": "SW Developer", "user_id": 7, "phone": "9991111" }
{ "name": "Henry", "lastname": "Carlson", "role": "Marketing", "user_id": 8, "phone": "1112222" }
```



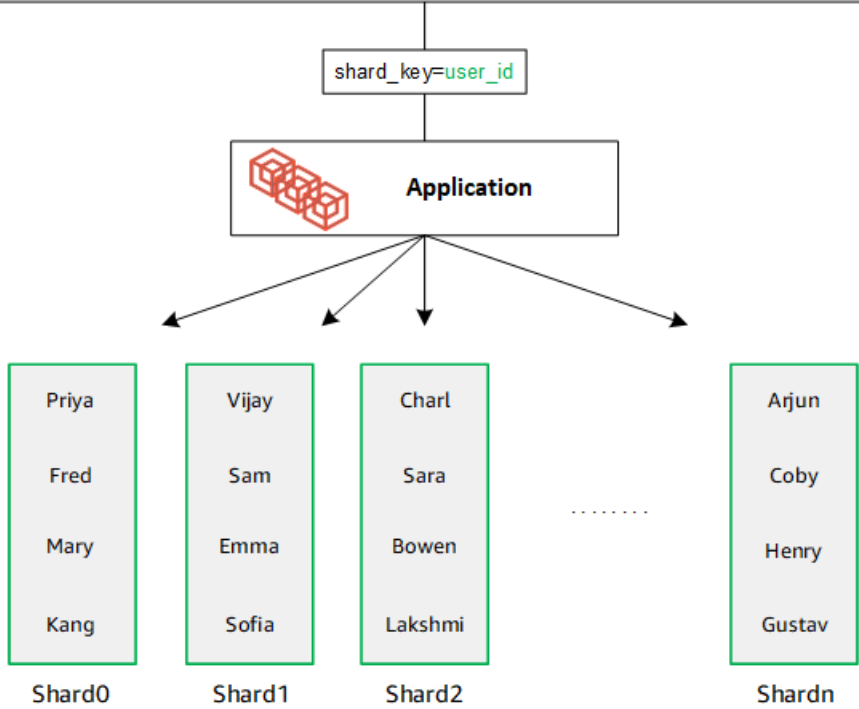
DocumentDB 使用散列分片在基礎碎片中對數據進行分區。其他數據的插入和分佈方式相同：

Employee Dataset

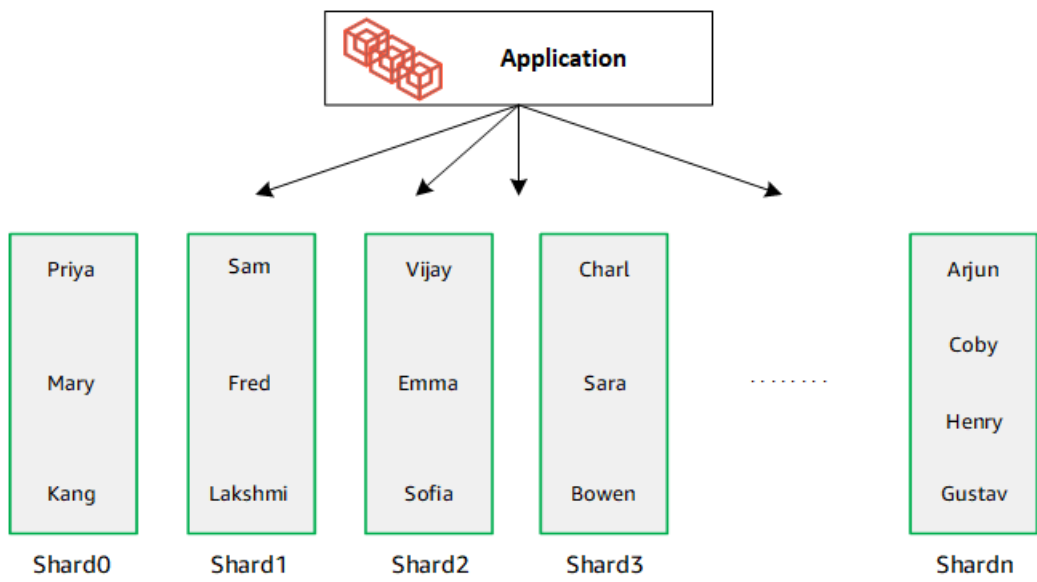
```

{"name": "Sam", "lastname": "Fender", "role": "Manager", "user_id": 9, "phone": "2223333"}
{"name": "Gustav", "lastname": "Friedrich", "role": "Manager", "user_id": 10, "phone": "3334444"}
{"name": "Sara", "lastname": "Goldstien", "role": "Manager", "user_id": 11, "phone": "4445555"}
{"name": "Fred", "lastname": "Williams", "role": "SW Architect", "user_id": 12, "phone": "6667777"}
{"name": "Sofia", "lastname": "Velez", "role": "SW Architect", "user_id": 13, "phone": "7778888"}
{"name": "Lakshmi", "lastname": "Ghosh", "role": "SW Developer", "user_id": 14, "phone": "8889999"}
{"name": "Coby", "lastname": "Jones", "role": "SW Developer", "user_id": 15, "phone": "9991111"}
{"name": "Kang", "lastname": "Zhu", "role": "Marketing", "user_id": 16, "phone": "1112222"}

```



當您透過新增其他碎片來擴展資料庫時，Amazon DocumentDB 會自動重新分配資料：



彈性叢集移轉

Amazon DocumentDB 支持將 MongoDB 的分片數據遷移到彈性叢集。支援離線、線上和混合式移轉方法。如需詳細資訊，請參閱 [遷移到 Amazon DocumentDB](#)。

彈性叢集擴展

Amazon DocumentDB 彈性叢集提供了增加彈性叢集中碎片數量 (向外擴充) 的能力，以及套用至每個碎片的 vCPUs 數量 (擴展)。您也可以視需要減少碎片和運算容量 (vCPUs) 的數量。

如需縮放最佳作法，請參閱 [擴展](#)。

Note

也可以使用叢集層級調整。如需詳細資訊，請參閱 [擴展 Amazon DocumentDB 叢集](#)。

彈性叢集可靠性

Amazon DocumentDB 的設計是為了可靠、耐用和容錯。為了提高可用性，彈性叢集會在不同可用區域中的每個碎片部署兩個節點。Amazon DocumentDB 包含數種自動功能，使其成為可靠的資料庫解決方案。如需詳細資訊，請參閱 [Amazon DocumentDB 可靠性](#)。

彈性叢集儲存與可用性

Amazon DocumentDB 資料存放在叢集磁碟區中，叢集磁碟區是使用固態硬碟 (SSD) 的單一虛擬磁碟區。叢集磁碟區由六份資料副本組成，這些副本會在單 AWS 一區域中跨多個可用區域自動複寫。此複寫有助於確保您的資料具有高耐用性，同時降低資料遺失的機率。它也有助於確保您的叢集在容錯移轉時提供更高的可用性，因為資料副本已存在於其他可用區域。如需儲存、高可用性和複寫的詳細資訊，請參閱 [Amazon DocumentDB: 它是如何工作的](#)。

Amazon DocumentDB 4.0 和彈性叢集之間的功能差異

Amazon DocumentDB 4.0 和彈性叢集之間存在下列功能差異。

- 來自 top 和 collStats 由碎片分區的結果。對於分片集合，數據分配在多個分區和 collScans 從分區匯總的 collStats 報告。
- 叢集碎片計數變更時，會重設來自 top 和 collStats 分片集合的集合統計資料。

- 備份內建角色現在支援serverStatus。動作-具備備份角色的開發人員和應用程式可以收集有關 Amazon DocumentDB 叢集狀態的統計資料。
- SecondaryDelaySecs欄位會slaveDelay在replSetGetConfig輸出中取代。
- 該hello命令替換 isMaster-hello 返回描述彈性集群的作用的文檔。
- 彈性叢集中的\$elemMatch運算子只會比對陣列第一個巢狀層級中的文件。在 Amazon DocumentDB 4.0 中，運營商在返回匹配的文檔之前遍歷所有級別。例如：

```

db.foo.insert(
[
  {a: {b: 5}},
  {a: {b: [5]}},
  {a: {b: [3, 7]}},
  {a: [{b: 5}]},
  {a: [{b: 3}, {b: 7}]},
  {a: [{b: [5]}]},
  {a: [{b: [3, 7]}]},
  {a: [[{b: 5}]]},
  {a: [[{b: 3}, {b: 7}]]},
  {a: [[{b: [5]}]]},
  {a: [[{b: [3, 7]}]]}
]);
// Elastic Clusters
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }

// Docdb 4.0: traverse more than one level deep
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }
{ "a" : [ [ { "b" : [ 5 ] } ] ] }

```

- Amazon DocumentDB 4.0 中的「\$」投影返回所有字段的所有文檔。使用彈性叢集時，具有「\$」投影的find命令會傳回符合僅包含符合「\$」投影之欄位的查詢參數的文件。
- 在彈性集群中，具有\$regex和\$options查詢參數的find命令返回錯誤：「無法在 \$ 正則表達式和 \$ 選項中設置選項」。
- 使用彈性叢集，\$indexOfCP現在會在下列情況傳回「-1」：

- 在中找不到子字串string expression , 或
- start是一個大於end , 或
- start是大於字串位元組長度的數字。

在 Amazon DocumentDB 4.0 , 當start位置是一個大於end或字符串的字節長度的數字\$indexOfCP返回「0」。

- 使用彈性叢集時_id fields , 中的投影作業 (例如 : {"_id.nestedField" : 1}) 會傳回僅包含投影欄位的文件。而在 Amazon DocumentDB 4.0 中 , 嵌套的字段投影命令不會過濾掉任何文檔。

開始使用 Amazon DocumentDB 彈性叢集

此入門章節將逐步說明如何建立和查詢您的第一個彈性叢集。有許多方法可以連接並開始使用彈性叢集。本指南使用[AWS Cloud9](#)一個基於 Web 的終端機來連接和查詢您的彈性叢集使用 mongo shell 直接 AWS Management Console從。

主題

- [設定](#)
- [步驟 1 : 建立彈性叢集](#)
- [步驟 2 : 建立 AWS Cloud9 環境](#)
- [第 3 步 : 安裝蒙戈外殼](#)
- [步驟 4 : Connect 到新的彈性叢集](#)
- [第 5 步 : 分片您的收藏; 插入和查詢數據](#)

設定

如果您想要透過建立與 Amazon EC2 執行個體的 SSH 連線 , 從本機電腦連線到 Amazon DocumentDB , 請參閱[使用 Amazon EC2 連線](#)。

必要條件

在建立第一個 Amazon DocumentDB 叢集之前 , 您必須執行下列動作 :

創建一個 Amazon Web Services (AWS) 帳戶

在您可以開始使用 Amazon DocumentDB 之前，您必須有一個 Amazon Web Services (AWS) 帳戶。該 AWS 帳戶是免費的。您僅需按使用的服務和資源量付費。

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為最佳安全實務，[將管理存取權指派給管理使用者](#)，並且僅使用根使用者來執行[需要根使用者存取權的任務](#)。

設定所需的 AWS Identity and Access Management (IAM) 許可。

存取管理 Amazon DocumentDB 資源 (例如叢集、執行個體和叢集參數群組) 需要 AWS 可用來驗證請求的登入資料。如需詳細資訊，請參閱 [Amazon DocumentDB 的 Identity and Access Management](#)。

1. 在的搜尋列中輸入 IAM AWS Management Console，然後在下拉式功能表中選取 IAM。
2. 進入 IAM 主控台後，從導覽窗格中選取 [使用者]。
3. 選擇您的用戶名。
4. 點擊按鈕添加權限。
5. 選取直接連接現有政策。
6. AmazonDocDBFullAccess在搜索欄中鍵入並在搜索結果中出現時選擇它。
7. 單擊底部顯示「下一步：查看」的藍色按鈕。
8. 點擊底部顯示「添加權限」的藍色按鈕。

創建一個 Amazon Virtual Private Cloud (Amazon VPC)

只有在您還沒有預設的 Amazon VPC 時，才需要執行此步驟。如果不這樣做，請完成 Amazon VPC 使用者指南中的 [Amazon VPC 入門](#) 步驟 1。這將需要不到五分鐘。

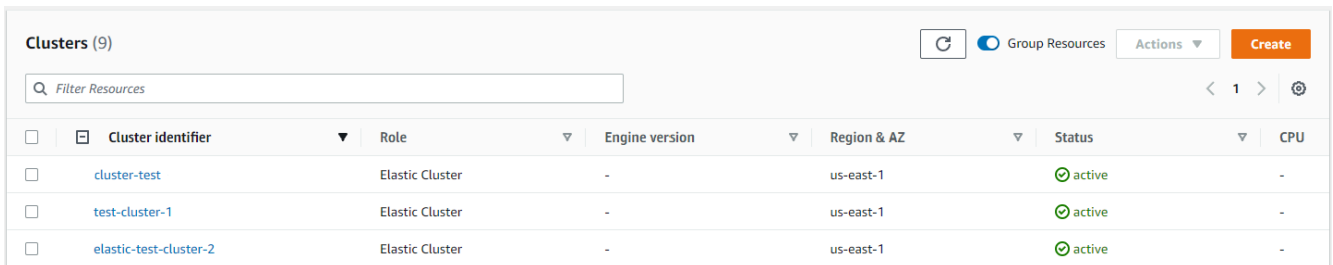
步驟 1：建立彈性叢集

在本節中，我們將說明如何使用 AWS Management Console 或 AWS CLI 與下列指示一起建立全新的彈性叢集。

Using the AWS Management Console

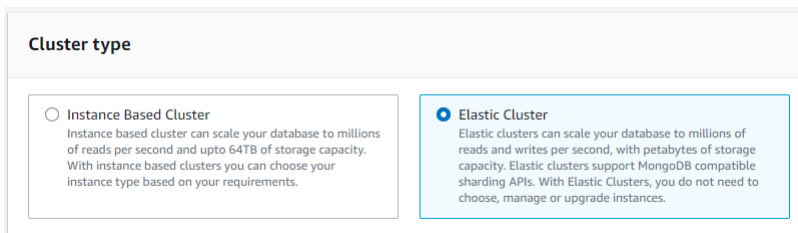
若要使用以下命令建立彈性叢集配置 AWS Management Console：

1. 登入 [AWS Management Console](#) 並開啟 Amazon DocumentDB 主控台。
2. 在 Amazon DocumentDB 理主控台的叢集下，選擇建立。



| Cluster identifier | Role | Engine version | Region & AZ | Status | CPU |
|------------------------|-----------------|----------------|-------------|--------|-----|
| cluster-test | Elastic Cluster | - | us-east-1 | active | - |
| test-cluster-1 | Elastic Cluster | - | us-east-1 | active | - |
| elastic-test-cluster-2 | Elastic Cluster | - | us-east-1 | active | - |

3. 在 [建立 Amazon DocumentDB 叢集] 頁面的 [叢集類型] 區段中，選擇彈性叢集。

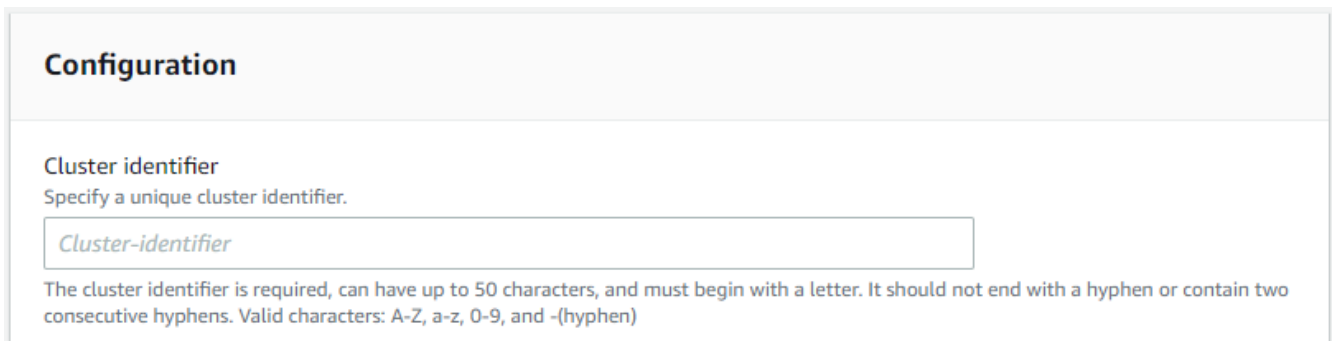


Cluster type

Instance Based Cluster
Instance based cluster can scale your database to millions of reads per second and upto 64TB of storage capacity. With instance based clusters you can choose your instance type based on your requirements.

Elastic Cluster
Elastic clusters can scale your database to millions of reads and writes per second, with petabytes of storage capacity. Elastic clusters support MongoDB compatible sharding APIs. With Elastic Clusters, you do not need to choose, manage or upgrade instances.

4. 在「建立 Amazon DocumentDB 叢集」頁面的「組態」區段中，輸入唯一的叢集識別碼 (遵循欄位下方的命名需求)。



Configuration

Cluster identifier
Specify a unique cluster identifier.

The cluster identifier is required, can have up to 50 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and -(hyphen)

5. 對於碎片配置字段：
 - a. 在 [碎片計數] 欄位中，輸入您想要在叢集中的碎片數目。每個叢集的碎片數目上限為 32。

Note

將為每個碎片部署兩個節點。兩個節點將具有相同的碎片容量。

- b. 在 [碎片執行個體計數] 欄位中，選擇要與每個碎片產生關聯的複本執行個體數目。碎片執行個體的最大數目為 16，以 1 為增量。所有複本執行個體都具有與下列欄位中定義的相同碎片容量。

Note

複本執行個體的數目會套用至彈性叢集中的所有碎片。碎片執行個體計數值為 1 表示有一個寫入器執行個體，而任何其他執行個體都是可用於讀取和改善可用性的複本。

- c. 在 [碎片容量] 欄位中，選擇要與每個碎片執行個體產生關聯的虛擬 CPU (vCPUs) 數目。每個碎片執行個體的 vCPUs 數目上限為 64 個。允許的值為 2、4、8、16、32、64。

Configuration

Cluster Name
Specify a unique cluster identifier.

The cluster identifier is required, can have up to 50 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and -(hyphen)

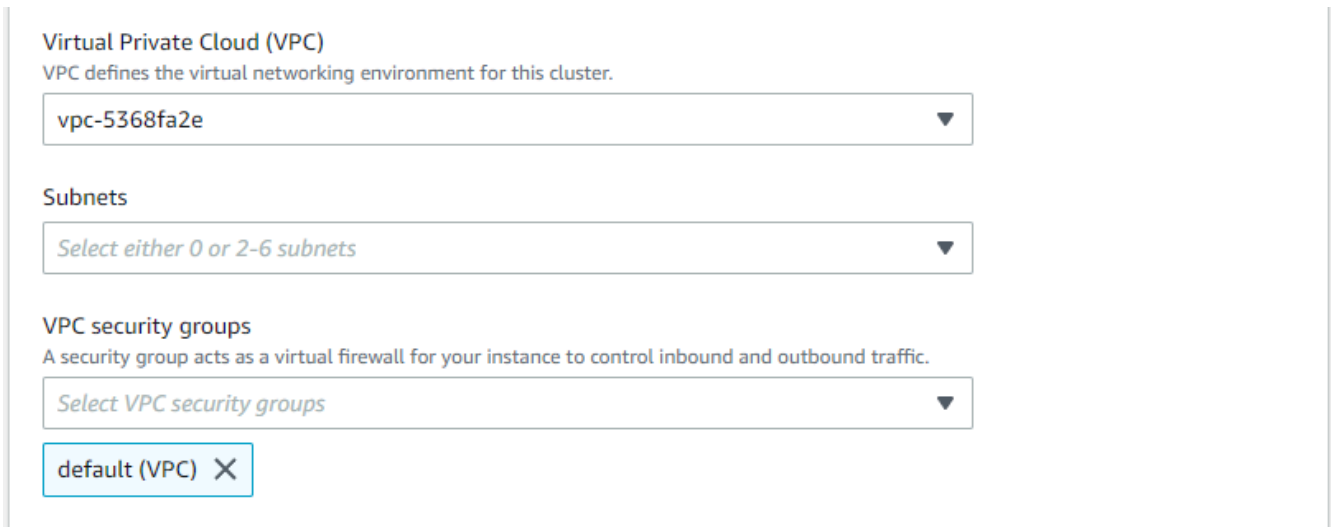
Shard count
Number of shards the Elastic Cluster will use.

Shard instance count
Number of instances for each shard. All instances will have the same shard capacity.

Shard capacity
vCPU capacity of each shard.

6. 在「Virtual Private Cloud (VPC)」欄位中，從下拉式清單中選擇 VPC。

對於子網路和 VPC 安全群組，您可以使用預設值或選取三個您選擇的子網路和最多三個 VPC 安全群組 (至少一個)。



Virtual Private Cloud (VPC)
VPC defines the virtual networking environment for this cluster.

vpc-5368fa2e ▼

Subnets
Select either 0 or 2-6 subnets ▼

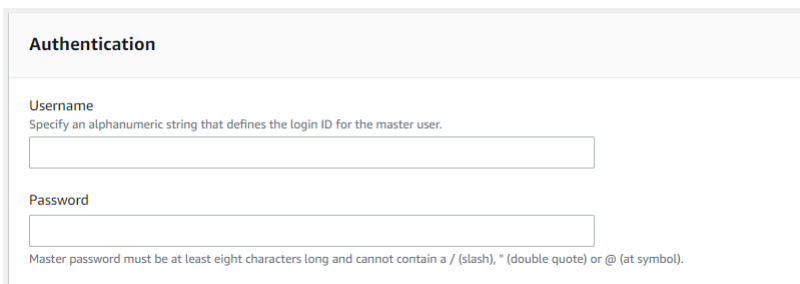
VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups ▼

default (VPC) X

7. 在「驗證」段落中，在「使用者名稱」欄位中輸入識別主要使用者登入名稱的字串。

在「密碼」欄位中，輸入符合指示的唯一密碼。



Authentication

Username
Specify an alphanumeric string that defines the login ID for the master user.

Password
Master password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).

8. 在「加密」區段中，保留預設設定。

或者，您可以輸入您建立的 AWS KMS key ARN。如需詳細資訊，請參閱 [Amazon DocumentDB Elastic Clusters 的靜態靜態資料加密](#)。

⚠ Important

彈性叢集必須啟用加密。

9. 在「Backup」區段中，根據您的備份需求編輯欄位。

Backup

Backup retention period
A period between 1 and 35 days in which automated backups are taken and retained.

1 day

Backup window
The daily time range (in UTC) during which automated backups are created.

Select window

No preference

- a. Backup 保留期間：在清單中，選擇在刪除叢集之前保留自動備份的天數。
- b. Backup 時段 — 設定 Amazon DocumentDB 要備份此叢集的每日時間和持續時間。
 - i. 如果您要設定建立備份的時間和持續時間，請選擇「選取視窗」。

開始時間 — 在第一個清單中，選擇開始時間小時 (UTC) 以開始自動備份。在第二個清單中，選擇您要自動備份開始進行的時間 (分)。

持續時間 — 在清單中，選擇要分配給建立自動備份的小時數。

- ii. 如果您希望 Amazon DocumentDB 選擇建立備份時的時間和持續時間，請選擇無偏好設定。

10. 在維護段落中，選擇將修改或修正程式套用至叢集的日期、時間和持續時間。

Maintenance

Maintenance window
The period in which pending modifications or patches are applied to instances in the cluster.

Select window

No preference

Start day: Monday

Start time: 00 : 00 UTC

Duration: 0.5 hours

11. 選擇 建立叢集。

現在已佈建彈性叢集。這可能需要幾分鐘才能完成。當彈性叢集狀態顯示為 [叢集] 清單 **active** 中時，您可以連線到叢集。

Using the AWS CLI

若要使用建立彈性叢集 AWS CLI，請搭配下列參數使用 `create-cluster` 作業：

- `--cluster-name` - 必要項目。在建立或上次修改期間輸入的彈性縮放叢集的目前名稱。
- `--shard-capacity` - 必要項目。指派給每個碎片的 vCPUs 數目。最大值為 64 個。允許的值为 2、4、8、16、32、64。

- `--shard-count` - 必要項目。指派給叢集的碎片數目。最大值為 32。
- `--shard-instance-count`選擇性。套用至此叢集中所有碎片的複本執行個體數目。最大值為 16。
- `--admin-user-name` - 必要項目。與管理員使用者關聯的使用者名稱。
- `--admin-user-password` - 必要項目。與管理員使用者相關聯的密碼。
- `--auth-type` - 必要項目。用來決定從何處擷取用於存取彈性叢集之密碼的驗證類型。有效類型為 `PLAIN_TEXT` 或 `SECRET_ARN`。
- `--vpc-security-group-ids`選擇性。設定要與此叢集關聯的 EC2 VPC 安全群組清單。
- `--preferred-maintenance-window`選擇性。以世界協調時間 (UTC) 為單位，設定可進行系統維護的每週時間範圍。

格式為：`ddd:hh24:mi-ddd:hh24:mi`。有效日期 (`ddd`)：星期一，星期二，星期三，星期四，星期五，星期六，星期日

預設值為每個 Amazon Web Services 區域的 8 小時時段隨機選取 30 分鐘的時段，每週的隨機日期發生。

最少 30 分鐘的窗口。

- `--kms-key-id`選擇性。設定加密叢集的 KMS 金鑰識別碼。

KMS 金鑰識別碼是 AWS KMS 加密金鑰的 Amazon 資源名稱 (ARN)。如果您使用擁有用於加密新叢集的 KMS 加密金鑰的相同 Amazon Web Services 帳戶建立叢集，則可以使用 KMS 金鑰別名而非 ARN 作為 KMS 加密金鑰。

如果未在中指定加密金鑰，`KmsKeyId` 且 `StorageEncrypted` 參數為真，Amazon DocumentDB 會使用您的預設加密金鑰。

- `--preferred-backup-window`選擇性。建立自動備份的每日偏好時間範圍。預設值為 30 分鐘的視窗，每 AWS 區域個視窗從 8 小時的時間段中隨機選取。
- `--backup-retention-period`選擇性。自動備份保留的天數。預設值為 1。
- `--storage-encrypted`選擇性。設定叢集是否已加密。
 - `--no-storage-encrypted`指定叢集未加密。
- `--subnet-ids`選擇性。設定網路子網路識別碼。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

Note

下列範例包括建立特定 KMS 金鑰。若要使用預設的 KMS 金鑰，請勿包含 `--kms-key-id` 參數。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic create-cluster \  
  --cluster-name sample-cluster-123 \  
  --shard-capacity 8 \  
  --shard-count 4 \  
  --shard-instance-count 3 \  
  --auth-type PLAIN_TEXT \  
  --admin-user-name testadmin \  
  --admin-user-password testPassword \  
  --vpc-security-group-ids ec-65f40350 \  
  --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \  
  --subnet-ids subnet-9253c6a3, subnet-9f1b5af9 \  
  --preferred-backup-window 18:00-18:30 \  
  --backup-retention-period 7
```

針對 Windows：

```
aws docdb-elastic create-cluster ^  
  --cluster-name sample-cluster-123 ^  
  --shard-capacity 8 ^  
  --shard-count 4 ^  
  --shard-instance-count 3 ^  
  --auth-type PLAIN_TEXT ^  
  --admin-user-name testadmin ^  
  --admin-user-password testPassword ^  
  --vpc-security-group-ids ec-65f40350 ^  
  --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^  
  --subnet-ids subnet-9253c6a3, subnet-9f1b5af9 \  
  --preferred-backup-window 18:00-18:30 \  
  --backup-retention-period 7
```

步驟 2：建立 AWS Cloud9 環境

AWS Cloud9 提供一個基於 Web 的終端機，您可以使用蒙戈殼層連接和查詢 Amazon DocumentDB 彈性叢集。

Note

注意：您的 AWS Cloud9 環境必須與執行個體位於相同的安全性群組中。您可以在 [Amazon EC2 主控台](#) 中變更安全群組。

1. 使用您的 AWS 帳戶並存取 AWS Management Console。
2. 導覽至主 AWS Cloud9 控制台。您可以在「搜尋」欄位中輸入「Cloud9」來尋找它。
3. 在 AWS Cloud9 環境首頁上，選擇建立環境。
4. 在名稱環境頁面的名稱欄位中，輸入您選擇的名稱。

選擇 下一個步驟。

Name environment

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.

testWebTerminalEnv

Limit: 60 characters

Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

Write a short description for your environment

Limit: 200 characters

Cancel Next step

5. 在 [環境類型] 區段下的 [環境設定] 中，選取 [為環境建立新的 EC2 執行個體 (直接存取)]。

在「執行個體類型」區段下，為您的網路選取適當的執行個體類型。

在「平台」區段下，選取 Amazon Linux 2 (建議使用)。

Configure settings

Environment settings

Environment type [Info](#)

Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports).

- Create a new EC2 instance for environment (direct access)**
Launch a new instance in this region that your environment can access directly via SSH.
- Create a new no-ingress EC2 instance for environment (access via Systems Manager)**
Launch a new instance in this region that your environment can access through Systems Manager.
- Create and run in remote server (SSH connection)**
Configure the secure connection to the remote server for your environment.

Instance type

- t2.micro (1 GiB RAM + 1 vCPU)**
Free-tier eligible. Ideal for educational users and exploration.
- t3.small (2 GiB RAM + 2 vCPU)**
Recommended for small-sized web projects.
- m5.large (8 GiB RAM + 2 vCPU)**
Recommended for production and general-purpose development.
- Other instance type**
Select an instance type.

t3.nano

Platform

- Amazon Linux 2 (recommended)**
- Amazon Linux AMI
- Ubuntu Server 18.04 LTS

6. 請展開 Network settings (advanced) (網路設定 (進階))。

選擇 VPC 和您在建立彈性叢集時使用的其中一個子網路。

選擇 下一個步驟。

▼ Network settings (advanced)

Network (VPC)
Launch your EC2 instance into an existing Amazon Virtual Private Cloud (VPC) or create a new one. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your new VPC.

vpc-5368fa2e (default) ↻ ↗ Create new VPC

Subnet
Select a public subnet in which the EC2 instance is created. (For a private subnet, you must create an environment that connects to its instance via Systems Manager.)

subnet-21a7eb00 | Default in us-east-1c ↻ ↗ Create new subnet

No tags associated with the resource.

Add new tag

You can add 50 more tags.

Cancel
Previous step
Next step

7. 檢閱您的 AWS Cloud9 組態。

如果您的組態正確，請選擇 [建立環境]。

第 3 步：安裝蒙戈外殼

一旦您的 AWS Cloud9 環境準備就緒，您就可以連線到叢集了。接下來，在您在步驟 3 中創建的 AWS Cloud9 環境中安裝 mongo 外殼。mongo shell 是一個命令行實用程序，用於連接和查詢彈性集群。

如果您的 AWS Cloud9 環境仍然從步驟 3 開啟，請返回該環境並跳至指令 3。如果您導覽離開您的 AWS Cloud9 環境，請在 AWS Cloud9 主控台的您的環境下，尋找標有您在上一個步驟中設定名稱的環境。選擇「開啟 IDE」。

1. 在命令提示字元中，使用下列命令建立儲存庫檔案：

Example

```
echo -e "[mongodb-org-4.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2013.03/mongodb-org/4.0/x86_64/\ngpgcheck=1 \nenabled=1
\ngpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-4.0.repo
```

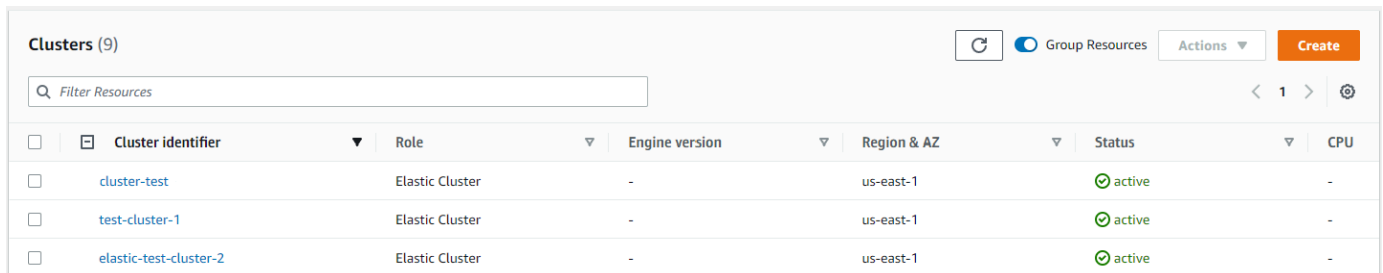
2. 完成後，請使用以下命令安裝 mongo 外殼：

```
sudo yum install -y mongodb-org-shell
```

步驟 4：Connect 到新的彈性叢集

使用您在步驟 4 中安裝的 mongo 外殼 Connect 到您的叢集。

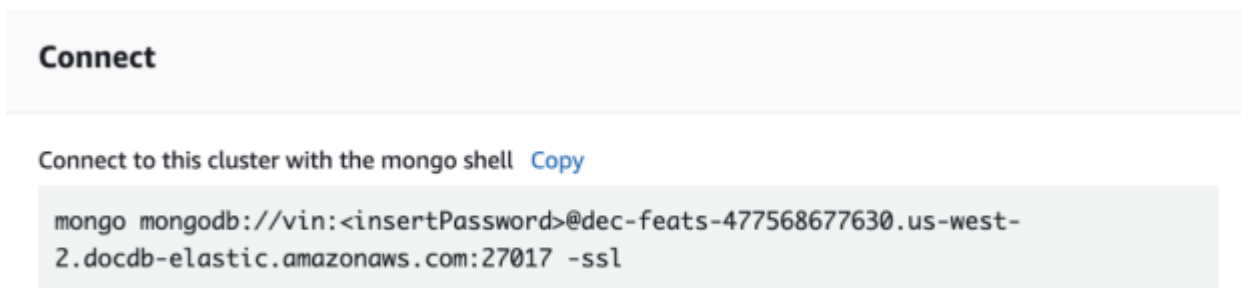
1. 在 Amazon DocumentDB 理主控台的叢集下，找到您的叢集。依角色排序，以顯示具有彈性叢集角色的所有叢集。



The screenshot shows the Amazon DocumentDB console interface. At the top, there's a header 'Clusters (9)' with a refresh button, a 'Group Resources' toggle, and an 'Actions' dropdown. Below the header is a search bar labeled 'Filter Resources'. The main content is a table with columns: Cluster identifier, Role, Engine version, Region & AZ, Status, and CPU. Three clusters are listed: 'cluster-test', 'test-cluster-1', and 'elastic-test-cluster-2', all with the role 'Elastic Cluster', engine version '-', region 'us-east-1', and status 'active'.

| Cluster identifier | Role | Engine version | Region & AZ | Status | CPU |
|------------------------|-----------------|----------------|-------------|--------|-----|
| cluster-test | Elastic Cluster | - | us-east-1 | active | - |
| test-cluster-1 | Elastic Cluster | - | us-east-1 | active | - |
| elastic-test-cluster-2 | Elastic Cluster | - | us-east-1 | active | - |

2. 選取叢集識別碼以選擇您建立的叢集。從連線和安全性，複製您的端點並將其貼到您的 AWS Cloud9 環境中。



3. 連接後，您應該會看到以下輸出：

```
Admin:~/environment $ mongo mongodb://vin:mytestpw@dec-feats-477568254530.us-west-2.docdb-elastic.amazonaws.com:27017 -ssl
MongoDB shell version v4.0.28
connecting to: mongodb://dec-feats-477568254530.us-west-2.docdb-elastic.amazonaws.com:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7413d0ae-43d4-426e-bbe8-c2dabb0b257b") }
MongoDB server version: 5.0.0
WARNING: shell and server versions do not match
mongos> █
```

第 5 步：分片您的收藏; 插入和查詢數據

彈性群集增加了對 Amazon DocumentDB 中的分片支持。現在您已連接到叢集，您可以分片叢集、插入資料並執行一些查詢。

1. 若要分片集合，請輸入下列內容：

```
sh.shardCollection("db.Employee1" , { "Employeeid" : "hashed" })
```

2. 若要插入單一文件，請輸入下列內容：

```
db.Employee1.insert({"Employeeid":1, "Name":"Joe", "LastName": "Bruin",  
"level": 1 })
```

將顯示以下輸出：

```
WriteResult({ "nInserted" : 1 })
```

3. 若要讀取您撰寫的文件，請輸入findOne()指令 (它會傳回單一文件)：

```
db.Employee1.findOne()
```

將顯示以下輸出：

Example

```
{  
  "_id" : ObjectId("61f344e0594fe1a1685a8151"),  
  "EmployeeID" : 1,  
  "Name" : "Joe",  
  "LastName" : "Bruin",  
  "level" : 1  
}
```

4. 若要執行更多查詢，請考慮使用遊戲設定檔使用案例。首先，將幾個條目插入到標題為「員工」的集合中。輸入下列資料：

Example

```
db.Employee1.insertMany([  
  { "Employeeid" : 1, "name" : "Matt", "lastname": "Winkle", "level": 12},  
  { "Employeeid" : 2, "name" : "Frank", "lastname": "Chen", "level": 2},  
  { "Employeeid" : 3, "name" : "Karen", "lastname": "William", "level": 7},  
  { "Employeeid" : 4, "name" : "Katie", "lastname": "Schaper", "level": 3}  
])
```

將顯示以下輸出：

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

5. 若要傳回設定檔集中的所有文件，請輸入 `find()` 指令：

```
db.Employee1.find()
```

顯示您在步驟 4 中輸入的資料。

6. 要查詢單個文檔，請包括一個過濾器（例如：「Katie」）。輸入下列資料：

```
db.Employee1.find({name: "Katie"})
```

將顯示以下輸出：

```
{ "_id" : 4, "name" : "Katie", "lastname": "Schaper", "level": 3 }
```

7. 若要尋找輪廓並加以修改，請輸入 `findAndModify` 指令。在這個例子中，員工「馬特」被賦予的「14」更高的水平：

Example

```
db.Employee1.findAndModify({
  query: { "Employeeid" : 1, "name" : "Matt"},
  update: { "Employeeid" : 1, "name" : "Matt", "lastname" : "Winkle", "level" :
    14 }
})
```

會顯示以下輸出（請注意，圖層尚未變更）：

Example

```
{
  "_id" : 1,
  "name" : "Matt",
  "lastname" : "Winkle",
  "level" : 12,
}
```

8. 若要驗證層級提升，請輸入下列查詢：

```
db.Employee1.find({name: "Matt"})
```

將顯示以下輸出：

```
{ "_id" : 1, "name" : "Matt", "lastname" : "winkle", "level" : 14 }
```

最佳實務

了解使 Amazon DocumentDB 叢集以[執行個體為基礎的 Amazon DocumentDB 叢集的所有最佳實務](#)也適用於彈性叢集。在新最佳實務確定時，會不斷更新此小節。

主題

- [選擇分片](#)
- [連線管理](#)
- [未分片的集合](#)
- [擴展](#)
- [監控彈性叢集](#)

選擇分片

下列清單說明建立碎片索引鍵的準則。

- 使用均勻分佈的散列鍵將數據分配到集群中的所有碎片（避免使用熱鍵）。
- 在所有讀取/更新/刪除請求中使用碎片密鑰，以避免散佈收集查詢。
- 執行讀取/更新/刪除操作時，請避免使用嵌套的碎片鍵。
- 進行批處理操作時，`ordered`請設置為 `false`，以便所有碎片都可以 `parallel` 運行並改善延遲。

連線管理

下列清單說明管理資料庫連線的準則。

- 監控您的連線計數，以及關閉新連線的開啟頻率。
- 將連線分配到應用程式組態中的所有子網路。如果您的叢集設定在多個子網路中，但您只使用子網路的子集，則可能會對最大連線造成瓶頸。

未分片的集合

以下描述了未分片集合的準則。

- 使用未分片集合時，要分配負載，請嘗試在不同的數據庫上保留高度利用的未分片集合。Amazon DocumentDB 彈性叢集會將資料庫放在不同的碎片之間，並將相同資料庫的未分割集合共放在相同的碎片上。

擴展

下列清單說明調整彈性叢集的準則。

- 擴展操作可能會造成短暫的資料庫和網路錯誤的間歇性。如果可能，請避免在尖峰時段縮放。嘗試在維護時段期間擴展。
- 與增加或減少碎片計數相比，首選擴展碎片容量（更改每個碎片的 vCPU 計數）以增加計算，因為它速度更快，並且間歇性數據庫和網絡錯誤的持續時間較短。
- 在預期增長時，傾向於增加碎片計數，而不是擴展碎片容量。這可讓您針對需要快速擴展的案例增加碎片容量來擴展叢集。
- 監控用戶端重試原則，並以指數輪詢和抖動重試，以避免在擴展時出現錯誤時資料庫過載。

監控彈性叢集

下列清單說明監視彈性叢集的準則。

- 追蹤每個碎片指標的 peak-to-average 比率，以判斷您是否要驅動非偶數流量（具有熱鍵/熱點）。追蹤 peak-to-average 比率的關鍵指標為：
 - PrimaryInstanceCPUUtilization
 - 這可以在每個碎片級別進行監視。
 - 在叢集層級，您可以監視 p99 偏斜的平均值。
 - PrimaryInstanceFreeableMemory
 - 這可以在每個碎片級別進行監視。
 - 在叢集層級，您可以監視 p99 偏斜的平均值。
 - DatabaseCursorsMax
 - 這應該在每個碎片層級進行監視，以確定偏斜。
 - Documents-Inserted/Updated/Returned/Deleted
 - 這應該在每個碎片層級進行監視，以確定偏斜。

管理彈性叢集

若要管理 Amazon 文件資料庫彈性叢集，您必須擁有具有適當 Amazon DocumentDB 控制平面許可的 IAM 政策。這些權限可讓您建立、修改和刪除叢集。Amazon DocumentDB FullAccess 策提供管理 Amazon 文件資料庫彈性叢集所需的所有許可。

下列主題說明如何在使用 Amazon DocumentDB 彈性叢集時執行各種任務。

主題

- [修改彈性叢集配置](#)
- [監視彈性叢集](#)
- [刪除彈性叢集](#)
- [管理彈性叢集快照](#)
- [停止和啟動 Amazon DocumentDB 彈性叢集](#)

修改彈性叢集配置

在本節中，我們將說明如何修改彈性叢集，使用 AWS Management Console 或 AWS CLI 與下列指示。

修改叢集的主要用途是透過增加或減少碎片計數和/或碎片運算容量來擴展碎片。

Using the AWS Management Console

若要使用以下命令修改彈性叢集配置 AWS Management Console：

1. 登入[AWS Management Console](#)並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在「叢集 ID」資料欄中選擇要修改的叢集名稱。
4. 選擇 Modify (修改)。
5. 編輯要變更的欄位，然後選取 [修改叢集]。

Configuration

Cluster identifier

SampleCluster

Shard count

Number of shards the Elastic Cluster will use.

2

Shard instance count

Number of instances for each shard. All instances will have the same shard capacity.

2

Shard capacity

vCPU capacity of each shard.

2

Maintenance

Maintenance window

The period in which pending modifications or patches are applied to your Elastic cluster.

- Select window
- No preference

Authentication

Username

SampleUser

New password

Confirm new password

Password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).

Network settings

Subnets

Select either 0 or 2-6 subnets

subnet-0b2962f92a0f5a8fb X

subnet-08c6d849efd4dfe96 X

VPC security groups

Note

或者，您也可以移至叢集頁面，核取叢集旁邊的方塊，選擇動作，然後選擇修改，以存取修改叢集對話方塊。

Using the AWS CLI

若要使用修改彈性叢集配置 AWS CLI，請使用具有下列參數的 `update-cluster` 作業：

- **--cluster-arn** - 必要項目。您要修改之叢集的 ARN 識別碼。
- **--shard-capacity** 可選。指派給每個碎片的 vCPUs 數目。最大值是 64 個。允許的值為 2、4、8、16、32、64。
- **--shard-count** 可選。指派給叢集的碎片數目。最大值為 32。
- **--shard-instance-計數**-可選。套用至此叢集中所有碎片的複本執行個體數目。最大值為 16。
- **--auth-type** 可選。用來決定從何處擷取用於存取彈性叢集之密碼的驗證類型。有效類型為 `PLAIN_TEXT` 或 `SECRET_ARN`。
- **--admin-user-password** 可選。與管理員使用者相關聯的密碼。
- **--vpc-security-group-ids** 可選。設定要與此叢集建立關聯的 Amazon EC2 和 Amazon 虛擬私有雲端 (VPC) 安全群組清單。
- **--preferred-maintenance-window** 可選。以世界協調時間 (UTC) 設定每週可進行系統維護的時間範圍

格式為：`ddd:hh24:mi-ddd:hh24:mi`。有效日期 (`ddd`)：星期一，星期二，星期三，星期四，星期五，星期六，星期日

預設值為每個 Amazon Web Services 區域的 8 小時時段隨機選取 30 分鐘的時段，每週的隨機日期發生。

最少 30 分鐘的窗口。

- **--subnet-ids** 可選。設定網路子網路識別碼。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic update-cluster \  
  --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \  
  --shard-capacity 8 \  
  --shard-count 4 \  
  --shard-instance-count 3 \  
  --admin-user-password testPassword \  
  --vpc-security-group-ids ec-65f40350 \  
  --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

針對 Windows :

```
aws docdb-elastic update-cluster ^  
  --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^  
  --shard-capacity 8 ^  
  --shard-count 4 ^  
  --shard-instance-count 3 ^  
  --admin-user-password testPassword ^  
  --vpc-security-group-ids ec-65f40350 ^  
  --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

若要在修改後監視彈性叢集的狀態，請參閱監視彈性叢集。

監視彈性叢集

在本節中，我們將說明如何使用 AWS Management Console 或 AWS CLI 以下指示來監視彈性叢集。

Using the AWS Management Console

若要使用以下命令監視彈性叢集配置 AWS Management Console：

1. 登入 [AWS Management Console](#) 並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在「叢集 ID」資料欄中選擇要監督的叢集名稱。

4. 選擇 Monitoring (監控) 索引標籤。

| ▼ Summary | | | |
|-------------------------------|--|------------------|---------------------------|
| Cluster Name SampleCluster | Cluster identifier cc05c8f6-e529-4f10-87d5-7ee3b5b4c7b9 | Shard count 2 | Shard capacity 2 vCPUs |
| Instances per shard 2 | Cluster status 🟢 active | | |

Connectivity & security | Configuration | Tags | **Monitoring**

以下監控類別會顯 CloudWatch 示來自 Amazon 的一些圖表：

- 資源使用率
- 輸送量
- Latency (延遲)
- 操作
- 系統

您也可以 CloudWatch 透過存取 Amazon，為彈性叢集設定自己的監控環境。AWS Management Console

Using the AWS CLI

若要使用監視特定的彈性叢集配置 AWS CLI，請搭配下列參數使用 `get-cluster` 作業：

- **--cluster-arn** - 必要項目。您想要資訊之叢集的 ARN 識別碼。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic get-cluster \
  --cluster-arn arn:aws:docdb-elastic:us-west-2:123456789012:cluster:/68ffcdf8-
e3af-40a3-91e4-24736f2dacc9
```

針對 Windows：

```
aws docdb-elastic get-cluster ^
```

```
--cluster-arn arn:aws:docdb:-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

此操作的輸出如下所示：

```
"cluster": {  
  ...  
  "clusterArn": "arn:aws:docdb-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9",  
  "clusterEndpoint": "stretch-11-477568257630.us-east-1.docdb-elastic.amazonaws.com",  
  "readerEndpoint": "stretch-11-477568257630-ro.us-east-1.docdb-elastic.amazonaws.com",  
  "clusterName": "stretch-11",  
  "shardCapacity": 2,  
  "shardCount": 3,  
  "shardInstanceCount": 5,  
  "status": "ACTIVE",  
  ...  
}
```

如需詳細資訊，請參閱 Amazon 文件資源管理 API 參考資料 `DescribeClusterSnapshot` 中的一文。

若要使用檢視所有彈性叢集的詳細資訊 AWS CLI，請使用具有下列參數的 `list-clusters` 作業：

- **--next-token** 可選。如果項目輸出數量 (`--max-results`) 少於基礎 API 呼叫傳回的項目總數，則輸出會包含一個 `NextToken`，提供您在後續命令中傳送以擷取下一組項目。
- **--max-results** 可選。要在命令輸出中傳回的總項目數。如果存在的結果多於指定 `max-results` 值，則響應中包含分頁令牌 (`next-token`)，以便可以檢索剩餘的結果。
 - 預設：100
 - 最少 20 人，最多 100 人

在下列範例中，將每個 `#####` 取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic list-clusters \  
  --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxZjQ== \  
  --cluster-arn arn:aws:docdb:-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

```
--max-results 2
```

針對 Windows :

```
aws docdb-elastic list-clusters ^  
--next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfg== ^  
--max-results 2
```

此操作的輸出如下所示 :

```
{  
  "Clusters": [  
    {  
      "ClusterIdentifier": "mycluster-1",  
      "ClusterArn": "arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster",  
      "Status": "available",  
      "ClusterEndpoint": "sample-cluster.sharded-cluster-corcjozrlsfc.us-west-2.docdb.amazonaws.com"  
    },  
    {  
      "ClusterIdentifier": "mycluster-2",  
      "ClusterArn": "arn:aws:docdb:us-west-2:987654321098:sharded-cluster:sample-cluster",  
      "Status": "available",  
      "ClusterEndpoint": "sample-cluster2.sharded-cluster-corcjozrlsfc.us-west-2.docdb.amazonaws.com"  
    }  
  ]  
}
```

刪除彈性叢集

在本節中，我們將說明如何刪除彈性叢集，使用 AWS Management Console 或 AWS CLI 與下列指示。

Using the AWS Management Console

若要使用以下命令刪除彈性叢集配置 AWS Management Console :

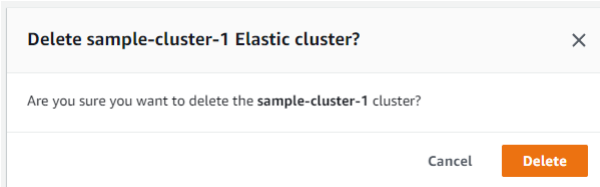
1. 登入 [AWS Management Console](#) 並開啟 Amazon DocumentDB 主控台。

2. 在導覽窗格中，選擇叢集。

i Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在叢集清單表格中，選取要刪除之叢集名稱左邊的核取方塊，然後選擇 [動作]。從下拉式選單中選擇 Delete (刪除)。
4. 在刪除「集群名稱」彈性集群中？」對話方塊中選擇「刪除」。



刪除叢集需要幾分鐘的時間。若要監控叢集的狀態，請參閱[監控 Amazon DocumentDB 叢集的狀態](#)。

Using the AWS CLI

若要使用刪除彈性叢集 AWS CLI，請使用具有下列參數的delete-cluster作業：

- **--cluster-arn** - 必要項目。您要刪除之叢集的 ARN 識別碼。
- **--no-skip-final-backup**可選。如果您想要最終備份，則必須將此參數加入最終備份的名稱。您必須包含 **--final-backup-identifier** 或 **--skip-final-backup**。
- **--skip-final-backup**可選。只有在您不想在刪除叢集之前進行最終備份時，才使用此參數。預設設定是建立最後的快照。

下列 AWS CLI 程式碼範例會刪除具有 ARN: awn: 文件資料庫:美國西部 -2:123456789012: 分割叢集:具有最終備份的範例叢集。

在下列範例中，將每個#####取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic delete-cluster \  
  --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster \  
  --no-skip-final-backup \  
  --final-backup-identifier sample-cluster-backup
```

```
--final-backup-identifier finalArnBU-arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster
```

針對 Windows :

```
aws docdb-elastic delete-cluster ^  
  --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster ^  
  --no-skip-final-backup ^  
  --final-backup-identifier finalArnBU-arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster
```

下列 AWS CLI 程式碼範例會刪除 ARN 為 ARN 的叢集:aw:docdb: 美國西部 -2:123456789012: 分割叢集:範例叢集而不進行最終備份。

在下列範例中，將每個#####取代為您自己的資訊。

若為 Linux、macOS 或 Unix :

```
aws docdb-elastic delete-cluster \  
  --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster \  
  --skip-final-backup \  
  --final-backup-identifier finalArnBU-arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster
```

針對 Windows :

```
aws docdb-elastic delete-cluster ^  
  --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster ^  
  --skip-final-backup ^
```

delete-cluster作業的輸出是您要刪除之叢集的顯示。

刪除叢集需要幾分鐘的時間。若要監控叢集的狀態，請參閱[監控 Amazon DocumentDB 叢集的狀態](#)。

管理彈性叢集快照

建立彈性叢集之後，即可建立手動快照。自動備份會在建立彈性叢集快照集時建立。

Note

您的彈性叢集必須 Available 處於狀態，才能擷取手動快照。

本節說明如何建立、檢視、還原及刪除彈性叢集快照。

下列主題說明如何在使用 Amazon DocumentDB 彈性叢集快照時執行各種任務。

主題

- [建立手動彈性叢集快照](#)
- [檢視彈性叢集快照](#)
- [從快照還原彈性叢集](#)
- [複製彈性叢集快照](#)
- [刪除彈性叢集快照](#)
- [管理彈性叢集快照自動備份](#)

建立手動彈性叢集快照

在本節中，我們將說明如何使用 AWS Management Console 或 AWS CLI 搭配下列指示來建立手動彈性叢集快照。

Using the AWS Management Console

若要使用建立手動彈性叢集快照 AWS Management Console：

1. 登入 [AWS Management Console](#) 並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇快照。

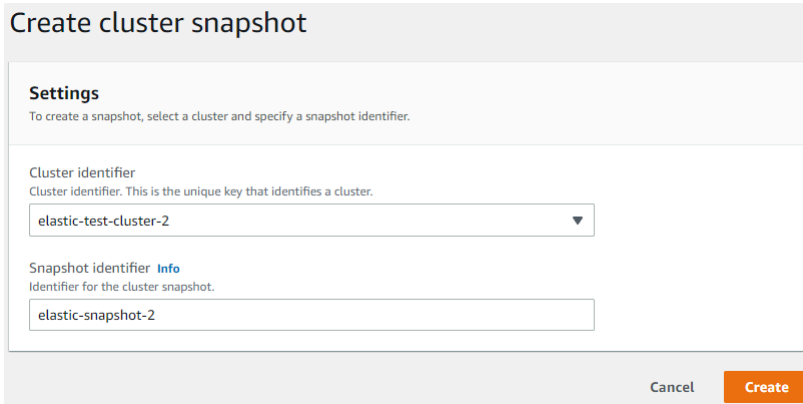
Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在 Snapshots (快照) 頁面選擇 Create (建立)。
4. 在 [建立叢集快照] 頁面的 [叢集識別碼] 欄位中，從下拉式清單中選擇您的彈性叢集。

在 [快照識別碼] 欄位中，輸入彈性叢集的唯一識別碼。

選擇建立。



Create cluster snapshot

Settings
To create a snapshot, select a cluster and specify a snapshot identifier.

Cluster identifier
Cluster identifier. This is the unique key that identifies a cluster.
elastic-test-cluster-2

Snapshot identifier [Info](#)
Identifier for the cluster snapshot.
elastic-snapshot-2

Cancel Create

Note

或者，您也可以移至 [叢集] 頁面，核取叢集旁的方塊，然後選擇動作，然後選取 [建立叢集快照]，以存取 [建立叢集快照] 對話方塊。

您的彈性叢集快照現已佈建。這可能需要幾分鐘才能完成。當快照清單中的狀態顯示時，您可以從快照 **Available** 中檢視和還原。

Using the AWS CLI

若要使用建立手動彈性叢集快照 AWS CLI，請使用具有下列參數的 `create-cluster-snapshot` 作業：

- **--snapshot-name** - 必要項目。您要建立的叢集快照名稱。
- **--cluster-arn** - 必要項目。您要建立快照集之叢集的 ARN 識別碼。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic create-cluster-snapshot \  
  --snapshot-name sample-snapshot-1 \  
  --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster
```

針對 Windows：

```
aws docdb-elastic create-cluster-snapshot ^
--snapshot-name sample-snapshot-1 ^
--cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster
```

檢視彈性叢集快照

在本節中，我們將說明如何使用 AWS Management Console 或 AWS CLI 與下列指示一起檢視彈性叢集快照資訊。

Using the AWS Management Console

若要使用以下命令檢視特定彈性叢集快照的相關資訊 AWS Management Console：


1. 登入 [AWS Management Console](#) 並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇快照。

Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在 [快照] 頁面上，按一下 [快照識別碼] 欄中的名稱，從清單中選擇您的快照。
4. 在詳細資料中檢視快照的資訊。

test-snapshot-id-1

| ▼ Details | |
|---|--|
| ARN arn:aws:rds:us-east-1:477568257630:cluster-snapshot:test-snapshot-id-1 | Snapshot identifier test-snapshot-id-1 |
| Cluster Name docdb-2022-07-18-22-22-13 | VPC vpc-5368fa2e |
| Snapshot type manual | Engine docdb |
| Engine version 4.0.0 | Master username vin |
| Status  available | Storage 6 GiB |
| Storage type manual | Snapshot creation time 10/25/2022, 4:02:04 PM UTC-5 |
| KMS key ID arn:aws:kms:us-east-1:477568257630:key/93644e8d-77ea-484c-80a6-8fb24c901385 | Cluster creation time 7/18/2022, 5:22:59 PM UTC-5 |

Using the AWS CLI

若要使用檢視有關特定彈性叢集快照的資訊 AWS CLI，請搭配下列參數使用 `get-cluster-snapshot` 作業：

- **--snapshot-arn** - 必要項目。您想要資訊之快照的 ARN 識別碼。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic get-cluster-snapshot \  
  --snapshot-arn sampleResourceName
```

針對 Windows：

```
aws docdb-elastic get-cluster-snapshot ^  
  --snapshot-arn sampleResourceName
```

若要使用檢視有關特定彈性叢集快照的資訊 AWS CLI，請搭配下列參數使用 `get-cluster-snapshot` 作業：

- **--snapshot-arn** - 必要項目。您想要資訊之快照的 ARN 識別碼。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic get-cluster-snapshot \  
  --snapshot-arn sampleResourceName
```

針對 Windows：

```
aws docdb-elastic get-cluster-snapshot ^  
  --snapshot-arn sampleResourceName
```

若要使用檢視所有彈性叢集快照的相關資訊 AWS CLI，請搭配下列參數使用 `list-cluster-snapshots` 作業：

- **--snapshot-type** 可選。要傳回的叢集快照類型。您可以指定下列其中一個值：
 - automated-傳回 Amazon DocumentDB 為您 AWS 的帳戶自動建立的所有叢集快照。
 - manual-傳回您為 AWS 帳戶手動建立的所有叢集快照。
 - shared-傳回已共用至您 AWS 帳戶的所有手動叢集快照。
 - public-傳回所有已標記為公用的叢集快照。
- **--next-token** 可選。前一個請求提供的選用分頁字符。如果指定了這個參數，則回應只會包含超出此 Token 的記錄，直到由指定的值為止max-results。
- **--max-results** 可選。要包含在回應中的結果數目上限。如果存在的結果多於指定max-results值，則響應中包含分頁令牌 (next-token)，以便可以檢索剩餘的結果。
 - 預設：100
 - 最少 20 人，最多 100 人

在下列範例中，將每個#####取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic list-cluster-snapshots \  
  --snapshot-type value \  
  --next-token value \  
  --max-results 50
```

針對 Windows：

```
aws docdb-elastic list-cluster-snapshots ^  
  --snapshot-type value ^  
  --next-token value ^  
  --max-results 50
```

從快照還原彈性叢集

在本節中，我們將說明如何使用或與下列指示— AWS CLI 起從快照還原彈性叢集。AWS Management Console

Using the AWS Management Console

若要使用以下命令從快照還原彈性叢集 AWS Management Console：

1. 登入[AWS Management Console](#)並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇快照。

i Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在 [快照識別碼] 資料行中，選擇要用來還原叢集的快照左邊的按鈕。
4. 選擇動作，然後選擇還原。

Restore snapshot

You are creating a new cluster from a source instance from a cluster snapshot. This new cluster will have the default cluster parameter group.

Configuration

Snapshot Name
The name for the snapshot.
test-snapshot-id-1

Cluster identifier [Info](#)
Specify a unique cluster identifier.

Instance class [Info](#)

2 vCPUs 16GiB RAM

Number of instances [Info](#)

5. 在 [還原快照] 頁面的 [叢集識別碼] 欄位中，輸入新叢集的名稱。

i Note

對於任何手動快照還原，您必須建立新叢集。

6. 在「Virtual Private Cloud (VPC)」欄位中，從下拉式清單中選擇 VPC。
7. 對於子網路和 VPC 安全群組，您可以使用預設值或選取三個您選擇的子網路，以及最多三個 VPC 安全群組 (至少一個)。
8. 如果您滿意此叢集組態，請選擇 Restore cluster (還原叢集)，等待叢集還原。

Using the AWS CLI

若要使用從快照還原彈性叢集 AWS CLI，請使用具有下列參數的 `restore-cluster-from-snapshot` 作業：

- **--cluster-name** - 必要項目。在建立或上次修改期間輸入的彈性叢集目前名稱。
- **--snapshot-arn** - 必要項目。用來還原叢集之快照的 ARN 識別碼。
- **--vpc-security-group-ids** 可選。要與叢集建立關聯的一或多個 Amazon EC2 和 Amazon 虛擬私有雲端 (VPC) 安全群組。
- **--kms-key-id** 可選。設定加密叢集的 KMS 金鑰識別碼。

KMS 金鑰識別碼是 AWS KMS 加密金鑰的 Amazon 資源名稱 (ARN)。如果您使用擁有用於加密新叢集的 KMS 加密金鑰的相同 Amazon Web Services 帳戶建立叢集，則可以使用 KMS 金鑰別名而非 ARN 作為 KMS 加密金鑰。

如果未在中指定加密金鑰，`KmsKeyId` 且 `StorageEncrypted` 參數為真，Amazon DocumentDB 會使用您的預設加密金鑰。

- **--subnet-ids** 可選。網路子網路識別碼。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic restore-cluster-from-snapshot \  
  --cluster-name elastic-sample-cluster \  
  --snapshot-arn sampleResourceName \  
  --vpc-security-group-ids value ec-65f40350 \  
  --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \  
  --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

針對 Windows：

```
aws docdb-elastic restore-cluster-from-snapshot ^  
  --cluster-name elastic-sample-cluster ^  
  --snapshot-arn sampleResourceName ^  
  --vpc-security-group-ids value ec-65f40350 ^  
  --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^  
  --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

複製彈性叢集快照

在 Amazon DocumentDB 中，您可以複製相同區域內和相同帳戶內的手動和自動彈性叢集快照。在本節中，我們將說明如何使用 AWS Management Console 或複製彈性叢集快照集 AWS CLI。

Using the AWS Management Console

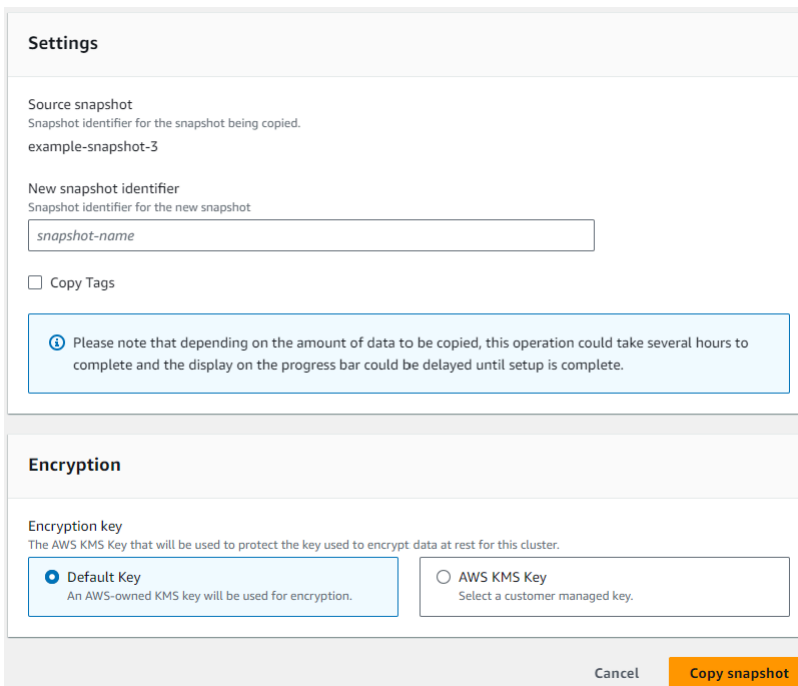
若要使用以下命令複製彈性叢集快照 AWS Management Console：

1. 登入 [AWS Management Console](#) 並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇快照。

Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在 [快照識別碼] 欄中，選擇您要複製的快照左側的按鈕。
4. 選擇動作，然後選擇複製。




Settings

Source snapshot
Snapshot identifier for the snapshot being copied.
example-snapshot-3

New snapshot identifier
Snapshot identifier for the new snapshot
snapshot-name

Copy Tags

 Please note that depending on the amount of data to be copied, this operation could take several hours to complete and the display on the progress bar could be delayed until setup is complete.

Encryption

Encryption key
The AWS KMS Key that will be used to protect the key used to encrypt data at rest for this cluster.

Default Key
An AWS-owned KMS key will be used for encryption.

AWS KMS Key
Select a customer managed key.

Cancel Copy snapshot

5. 在 [新增快照識別碼] 中，輸入新快照的名稱。
6. 若要將來源彈性叢集快照中的所有標籤複製到目標彈性叢集快照，請核取此方塊。
7. 對於加密，請選擇預設 AWS KMS 金鑰或您選擇的 KMS 金鑰。第二個選項可讓您選取已建立的現有 KMS 金鑰，或允許您建立新的 KMS 金鑰。

8. 完成時選擇 [複製快照]。

Using the AWS CLI

若要使用複製彈性叢集快照 AWS CLI，請搭配下列參數使用 `copy-cluster-snapshot` 作業：

- **--source-db-cluster-snapshot-identifier** - 必要項目。要複製之現有彈性叢集快照的識別碼。彈性叢集快照必須存在且處於可用狀態。如果您要將快照複製到另一個快照 AWS 區域，則此識別碼必須為來源 AWS 區域的 ARN 格式。此參數不區分大小寫。
- **--target-db-cluster-snapshot-identifier** - 必要項目。要從現有叢集快照建立之新彈性叢集快照的識別碼。此參數不區分大小寫。

目標快照名稱限制：

- 不能是現有快照的名稱。
- 長度為 [1—63] 字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

在下列範例中，將每個 `#####` 取代之為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic copy-cluster-snapshot \  
  --source-cluster-snapshot-arn <sample ARN> \  
  --target-cluster-snapshot-name my-target-copied-snapshot
```

針對 Windows：

```
aws docdb-elastic copy-cluster-snapshot ^  
  --source-cluster-snapshot-arn <sample ARN> ^  
  --target-cluster-snapshot-name my-target-copied-snapshot
```

刪除彈性叢集快照

在本節中，我們將說明如何使用 AWS Management Console 或刪除彈性叢集快照集 AWS CLI。

Using the AWS Management Console

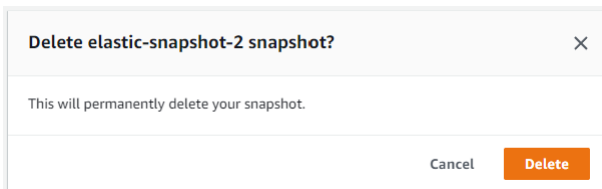
若要使用以下命令從快照還原彈性叢集 AWS Management Console：

1. 登入[AWS Management Console](#)並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇快照。

Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在 [快照識別碼] 資料行中，選擇要用來還原叢集的快照左邊的按鈕。
4. 依序選擇動作和刪除。



5. 在 [刪除快照名稱] 快照對話方塊中，選擇 [刪除]。

Using the AWS CLI

若要使用刪除彈性叢集快照 AWS CLI，請搭配下列參數使用 `delete-cluster-snapshot` 作業：

- **--snapshot-arn** - 必要項目。用來還原叢集之快照的 ARN 識別碼。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic delete-cluster-snapshot \  
  --snapshot-arn sampleResourceName
```

針對 Windows：

```
aws docdb-elastic delete-cluster-snapshot ^  
  --snapshot-arn sampleResourceName
```

管理彈性叢集快照自動備份

Amazon DocumentDB 會為您的彈性叢集拍攝每日快照。您可以在新的或現有的彈性叢集快照組態中，指定偏好的備份時段和備份保留期間。在本節中，我們將說明如何使用 AWS Management Console 或在彈性叢集快照中設定自動備份參數 AWS CLI。

Using the AWS Management Console

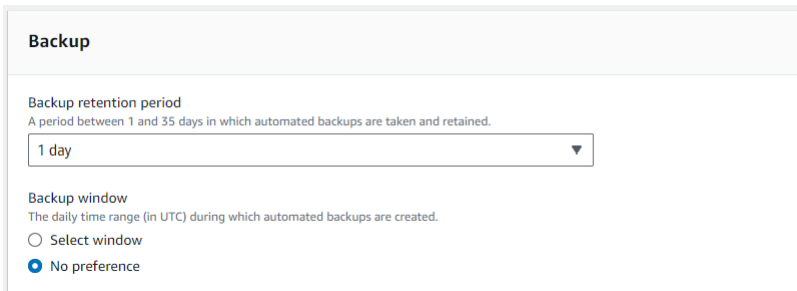
若要使用以下命令設定新彈性叢集快照的自動備份 AWS Management Console：

1. 登入 [AWS Management Console](#) 並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。

Tip

如果畫面左側沒有看到導覽窗格，請選擇導覽窗格左上角的選單圖示。

3. 在叢集識別碼資料欄中，選擇您要變更其備份設定值的叢集左邊的按鈕。
4. 選擇動作，然後選擇修改。
5. 在「Backup」區段中，根據您的備份需求編輯欄位。



Backup

Backup retention period
A period between 1 and 35 days in which automated backups are taken and retained.

1 day

Backup window
The daily time range (in UTC) during which automated backups are created.

Select window

No preference

- a. Backup 保留期間：在清單中，選擇在刪除叢集之前保留自動備份的天數。
- b. Backup 時段 — 設定 Amazon DocumentDB 要備份此叢集的每日時間和持續時間。
 - i. 如果您要設定建立備份的時間和持續時間，請選擇「選取視窗」。

開始時間 — 在第一個清單中，選擇開始時間小時 (UTC) 以開始自動備份。在第二個清單中，選擇您要自動備份開始進行的時間 (分)。

持續時間 — 在清單中，選擇要分配給建立自動備份的小時數。

- ii. 如果您希望 Amazon DocumentDB 選擇建立備份時的時間和持續時間，請選擇無偏好設定。

6. 完成時選擇 [修改叢集]。

Using the AWS CLI

若要使用設定新彈性叢集快照的自動備份 AWS CLI，請使用具有下列參數的 `create-cluster-snapshot` 作業：

- **--preferred-backup-window** 可選。建立自動備份的每日偏好時間範圍。預設值為 30 分鐘的視窗，每 AWS 區域個視窗從 8 小時的時間段中隨機選取。

約束：

- 格式必須為 `hh24:mi-hh24:mi`。
- 必須以國際標準時間 (UTC) 表示。
- 不得和慣用的維護時段衝突。
- 必須至少 30 分鐘。
- **--backup-retention-period** 可選。自動備份保留的天數。預設值為 1。

約束：

- 必須指定最小值 1。
- 範圍是從 1 到 35。

Note

只有當叢集處於「使用中」狀態時，才會執行自動備份。

Note

您也可以使用 `aws docdb-elastic update-cluster` 指令修改現有彈性叢集的 `preferred-backup-window` 和 `backup-retention-period` 參數。

在下列範例中，將每個 `#####` 取代為您自己的資訊。

```
##create-cluster##### Amazon DocumentDB ##### 7 #####
##### 18:00-18:30#
```

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic create-cluster \  
  --cluster-name sample-cluster \  
  --shard-capacity 2 \  
  --shard-count 2 \  
  --admin-user-name SampleAdmin \  
  --auth-type PLAIN_TEXT \  
  --admin-user-password SamplePass123! \  
  --preferred-backup-window 18:00-18:30 \  
  --backup-retention-period 7
```

針對 Windows：

```
aws docdb-elastic create-cluster ^  
  --cluster-name sample-cluster ^  
  --shard-capacity 2 ^  
  --shard-count 2 ^  
  --admin-user-name SampleAdmin ^  
  --auth-type PLAIN_TEXT ^  
  --admin-user-password SamplePass123! ^  
  --preferred-backup-window 18:00-18:30 ^  
  --backup-retention-period 7
```

停止和啟動 Amazon DocumentDB 彈性叢集

停止和啟動 Amazon DocumentDB 彈性叢集可協助您管理開發和測試環境的成本。您可以在不需要時暫時停止叢集，而不是每次使用 Amazon DocumentDB 時建立和刪除彈性叢集。然後，您可以在恢復測試時再次啟動它。

主題

- [停止及啟動彈性叢集概觀](#)
- [您可以在已停止的彈性叢集上執行的作業](#)

停止及啟動彈性叢集概觀

在您不需要 Amazon DocumentDB 彈性叢集的期間，您可以停止叢集。一旦您需要叢集，即可隨時重新啟動它。啟動和停止可簡化彈性叢集的設定和拆卸程序，這些叢集用於開發、測試或不需要持續可用

性的類似活動。您可以使用 AWS Management Console 或以單一動作來停止和啟動彈性叢集。AWS CLI

當您的彈性叢集停止時，叢集儲存磁碟區會維持不變。您只需支付指定的保留時段內儲存、手動快照和自動備份儲存的費用。Amazon DocumentDB 會在七天後自動啟動彈性叢集，因此不會落後於任何必要的維護更新之後。當您的叢集在七天後啟動時，將開始支付再次使用彈性叢集的費用。叢集停止時，您無法查詢儲存體磁碟區，因為查詢要求叢集處於可用狀態。

當 Amazon DocumentDB 彈性叢集停止時，就無法以任何方式修改叢集。這包括刪除叢集。

Using the AWS Management Console

下列程序說明如何停止處於可用狀態的彈性叢集，或啟動已停止的彈性叢集。

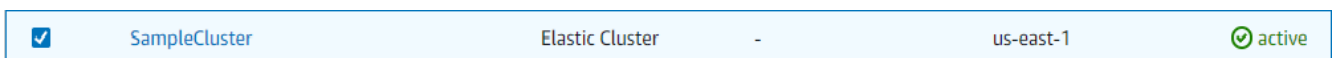
若要停止或啟動 Amazon DocumentDB 彈性叢集

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在導覽窗格中，選擇叢集。

Tip

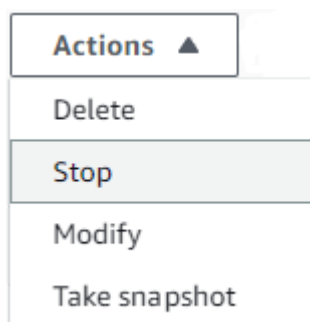
如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在叢集清單中，選擇您要停止或啟動之叢集名稱左側的按鈕。

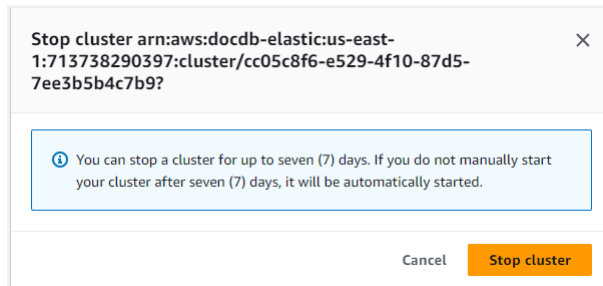


4. 選擇 Actions (動作)，然後選擇您想要在叢集上執行的動作。

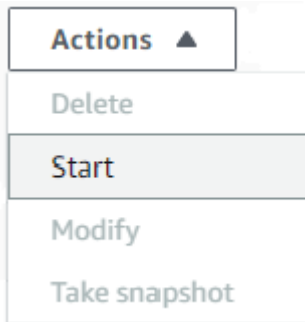
- 如果您想要停止可用的叢集：
 - a. 選擇停止。



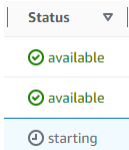
- b. 在確認對話方塊中，選擇停止叢集來確認您要停止彈性叢集，或選擇取消，以保持叢集持續執行。



- 如果您想要啟動叢集，而叢集已停止，請選擇 Start (啟動)。



5. 監視彈性叢集的狀態。如果您啟動了叢集，則可以在叢集可用時繼續使用叢集。如需詳細資訊，請參閱 [決定叢集的狀態](#)。



Using the AWS CLI

下列程式碼範例說明如何停止處於作用中或可用狀態的彈性叢集，或啟動已停止的彈性叢集。

若要使用停止彈性叢集 AWS CLI，請使用 `stop-cluster` 作業。若要啟動已停止的叢集，請使用 `start-cluster` 操作。這兩個操作均使用 `--cluster-arn` 參數。

參數：

- **--cluster-arn** - 必要項目。您要停止或啟動之彈性叢集的 ARN 識別碼。

Example — 若要使用停止彈性叢集 AWS CLI

在下列範例中，將每個 `#####` 取代為您自己的資訊。

下列程式碼會停止使用 ARN 的 `arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2` 彈性叢集。

Note

彈性叢集必須處於作用中或可用狀態。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic stop-cluster \  
  --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

針對 Windows：

```
aws docdb-elastic stop-cluster ^  
  --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Example — 若要使用啟動彈性叢集 AWS CLI

在下列範例中，將每個 `#####` 取代為您自己的資訊。

下列程式碼會以的 `arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2` ARN 啟動彈性叢集。

Note

彈性叢集目前必須停止。

若為 Linux、macOS 或 Unix：

```
aws docdb-elastic start-cluster \  
  --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

針對 Windows：

```
aws docdb-elastic start-cluster ^
```

```
--cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/  
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

您可以在已停止的彈性叢集上執行的作業

叢集停止時，您無法修改 Amazon DocumentDB 彈性叢集的組態。您必須先啟動叢集，然後才能執行任何這類管理動作。

Amazon DocumentDB 只會在已停止的彈性叢集再次啟動之後，才會將任何排定的維護套用至該叢集。在七天之後，Amazon DocumentDB 會自動啟動已停止的彈性叢集，這樣它就不會落後於其維護狀態太遠。當彈性叢集重新啟動時，您將開始再次支付叢集中的碎片費用。

當彈性叢集停止時，Amazon DocumentDB 不會執行任何自動備份，也不會延長備份保留期。

Amazon DocumentDB Elastic Clusters 的靜態靜態資料加密

下列主題可協助您了解、建立和監控 Amazon DocumentDB 彈性叢集的 AWS Key Management Service 加密金鑰：

主題

- [Amazon DocumentDB 彈性叢集如何使用授權 AWS KMS](#)
- [建立客戶受管金鑰受管](#)
- [監控 Amazon DocumentDB Elastic Clusters Clusters 的加密金鑰，以](#)
- [進一步了解](#)

Amazon DocumentDB 彈性叢集會自動與 AWS Key Management Service (AWS KMS) 整合以進行金鑰管理，並使用稱為包絡加密的方法來保護您的資料。如需封套加密的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [封套加密](#)。

AWS KMS key 是加密金鑰的邏輯表示。KMS 金鑰包含金鑰 ID、建立日期、說明和金鑰狀態等中繼資料。KMS 金鑰也包含可用來加密和解密資料的金鑰材料。如需 KMS 金鑰的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [AWS KMS keys](#)。

Amazon DocumentDB 彈性叢集支援使用兩種金鑰類型的加密：

- AWS 擁有的金鑰 — Amazon DocumentDB 彈性叢集預設會使用這些金鑰來自動加密個人識別資料。您無法檢視、管理或使用 AWS-擁有的金鑰，或稽核其使用方式。但是，您不必採取任何動作或變

更任何程式，即可保護加密您資料的金鑰，即可保護加密您的資料的金鑰，如需詳細資訊，請參閱AWS Key Management Service開發人員指南中的[AWS擁有金鑰](#)。

- 客戶受管金鑰 AWS KMS keys — 您建立、擁有和管理的對稱。由於您可以完全控制此加密層，因此您可以執行以下工作：
 - 建立和維護關鍵政策
 - 建立和維護 IAM 政策和撥款
 - 啟用和停用金鑰政策
 - 輪換金鑰加密材料
 - 新增標籤
 - 建立金鑰別名
 - 排程要刪除的金鑰

如需詳細資訊，請參閱AWS Key Management Service開發人員指南中的[客戶管理金鑰](#)。

Important

您必須使用對稱加密 KMS 金鑰，因為 Amazon DocumentDB 僅支援對稱加密 KMS 金鑰，因為 Amazon DocumentDB 僅支援對稱加密 KMS 金鑰，因請勿使用非對稱 KMS 金鑰來嘗試加密 Amazon DocumentDB Elastic Clusters Elastic Clusters 中的資料加密。如需詳細資訊，請參閱AWS Key Management Service開發人員指南AWS KMS中的[非對稱金鑰](#)。

如果 Amazon DocumentDB 再也無法存取叢集加密金鑰的存取 — 例如，當撤銷對金鑰的存取 — 加密的叢集會進入終止狀態。在此情況下，您只能從備份中還原叢集。對於 Amazon DocumentDB，備份始終啟用 1 天。此外，如果停用加密 Amazon DocumentDB 叢集的金鑰，您最終將失去該叢集的讀取和寫入存取權。當 Amazon DocumentDB 遇到經過金鑰加密的叢集，但 Amazon DocumentDB 會使叢集進入終止狀態。在此情況下，該叢集再也無法使用，而且無法復原資料庫的目前狀態。若要還原叢集，您必須重新啟用對 Amazon DocumentDB 加密金鑰的存取，然後從備份中還原叢集的存取。

Important

當您建立加密叢集的 KMS 金鑰在建立加密叢集後，您就無法變更它的 KMS 金鑰，請務必在建立加密的彈性叢集前，先決定您的加密金鑰要求要求。

Amazon DocumentDB 彈性集群如何使用授權 AWS KMS

Amazon DocumentDB Elastic Clusters 需要[授予](#)，才能使用您的客戶受管金鑰，以使用您

當您建立使用客戶受管金鑰加密的叢集，Amazon DocumentDB Elastic Clusters 會將CreateGrant請求傳送給，以代表您建立授予。AWS KMS中的授予AWS KMS是用來授予 Amazon DocumentDB Elastic Clusters 對客戶帳戶中 KMS 金鑰的存取權限。

Amazon DocumentDB Elastic Clusters 需要授予，才能在下列內部操作中使用客戶受管金鑰受管金鑰進行個體：

- 傳送DescribeKeyAWS KMS要求以確認建立追蹤器或地理圍欄集合時輸入的對稱客戶管理 KMS 金鑰 ID 是否有效。
- 傳送GenerateDataKey要求AWS KMS以產生由客戶管理金鑰加密的資料金鑰。
- 將Decrypt請求傳送AWS KMS至，以解密加密的資料金鑰，如此就能將這些金鑰傳送給，以便將這些金鑰傳送給
- 您可以隨時撤銷授予的存取權，或移除服務對客戶受管金鑰的存取權。如果您這麼做，Amazon DocumentDB Elastic Clusters 將無法存取由客戶受管金鑰加密的任何資料，這會影響與該資料有關的操作。

建立客戶受管金鑰受管

您可以使用或 API 來建立對稱的客戶受管金鑰受管金鑰，使用AWS Management Console或 AWS KMS API 來建立對稱

對稱式客戶管理金鑰建立

按照AWS Key Management Service開發人員指南中有關[創建對稱客戶管理密鑰](#)的步驟進行操作。

金鑰政策

關鍵原則可控制對客戶管理金鑰的存取。每個客戶受管金鑰必須只有一個金鑰政策，其中包含決定誰可以使用金鑰的陳述式，以及可以使用金鑰的使用方式。當您建立客戶受管金鑰時，您可以指定金鑰政策。如需詳細資訊，請參閱AWS Key Management Service開發人員指南AWS Key Management Service[概觀](#)中的 KMS 金鑰存取資訊。

若要將您的客戶受管金鑰與 Amazon DocumentDB 彈性叢集資源搭配使用，必須在金鑰政策中允許下列 API 操作：

- [kms:CreateGrant](#)— 將授權新增至客戶管理的金鑰。授予對指定 KMS 金鑰的控制存取權，以便授予 Amazon Location Service 所需的操作的存取權。如需有關使用授權的詳細資訊，請參閱AWS Key Management Service開發人員指南AWS KMS中的授權。
- [kms:DescribeKey](#)— 提供客戶管理的密鑰詳細信息，以允許 Docdb 彈性驗證密鑰。
- [kms:Decrypt](#)— 允許文檔庫彈性使用存儲的加密數據密鑰訪問加密數據。
- [kms:GenerateDataKey](#)— 允許 Docdb Elastic 生成加密的數據密鑰並將其存儲，因為數據密鑰不會立即用於加密。

如需詳細資訊，請參閱[金鑰原則中的AWS服務權限](#)和AWS Key Management Service開發人員指南中的[金鑰存取疑難排解](#)。

透過 IAM 政策限制客戶受管金鑰存取

除了 KMS 金鑰政策之外，您還可以在 IAM 政策中限制 KMS 金鑰許可。

您可以透過各種方式使 IAM 政策更加嚴格。例如，若要允許客戶受管金鑰僅用於源自 Amazon DocumentDB Elastic Clusters 的請求，您可以使用[kms:ViaService條件金鑰搭配值](#)。docdb-elastic.<region-name>.amazonaws.com

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的[允許其他帳戶中的使用者使用 KMS 金鑰](#)。

監控 Amazon DocumentDB Elastic Clusters Clusters 的加密金鑰，以

當您將AWS KMS key客戶託管密鑰與文檔庫彈性資源一起使用時，可以使用AWS CloudTrail或亞馬遜 CloudWatch日誌來跟踪文檔庫彈性發送到的請求。AWS KMS

下列範例是 Amazon DocumentDB 彈性叢集呼叫的AWS KMS key操作

GenerateDataKeyWithoutPlainTextDecrypt，DescribeKey以存取由客戶受管金鑰加密的資料所呼叫的、和監控AWS CloudTrail事件：CreateGrant

CreateGrant

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
```

```

    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE",
        "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Sampleuser01"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2023-05-09T23:04:20Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "docdb-elastic.amazonaws.com"
  },
  "eventTime": "2023-05-09T23:55:48Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "docdb-elastic.amazonaws.com",
  "userAgent": "docdb-elastic.amazonaws.com",
  "requestParameters": {
    "retiringPrincipal": "docdb-elastic.us-east-1.amazonaws.com",
    "granteePrincipal": "docdb-elastic.us-east-1.amazonaws.com",
    "operations": [
      "Decrypt",
      "Encrypt",
      "GenerateDataKey",
      "GenerateDataKeyWithoutPlaintext",
      "ReEncryptFrom",
      "ReEncryptTo",
      "CreateGrant",
      "RetireGrant",
      "DescribeKey"
    ],
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "responseElements": {
    "grantId":
      "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",

```

```

    "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "AWS Internal",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

GenerateDataKey

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE",
        "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Sampleuser01"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2023-05-10T18:02:59Z",
        "mfaAuthenticated": "false"
      }
    }
  }
}

```

```

    },
    "invokedBy": "docdb-elastic.amazonaws.com"
  },
  "eventTime": "2023-05-10T18:03:25Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "docdb-elastic.amazonaws.com",
  "userAgent": "docdb-elastic.amazonaws.com",
  "requestParameters": {
    "keySpec": "AES_256",
    "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "AWS Internal",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

Decrypt

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {

```

```
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AROAIKDTESTANDEXAMPLE",
      "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
      "accountId": "111122223333",
      "userName": "Sampleuser01"
    },
    "webIdFederationData": {},
    "attributes": {
      "creationDate": "2023-05-10T18:05:49Z",
      "mfaAuthenticated": "false"
    }
  },
  "invokedBy": "docdb-elastic.amazonaws.com"
},
"eventTime": "2023-05-10T18:06:19Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-east-1",
"sourceIPAddress": "docdb-elastic.amazonaws.com",
"userAgent": "docdb-elastic.amazonaws.com",
"requestParameters": {
  "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
  {
    "accountId": "AWS Internal",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}
```

DescribeKey

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE",
        "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Sampleuser01"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2023-05-09T23:04:20Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "docdb-elastic.amazonaws.com"
  },
  "eventTime": "2023-05-09T23:55:48Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "docdb-elastic.amazonaws.com",
  "userAgent": "docdb-elastic.amazonaws.com",
  "requestParameters": {
    "keyId": "alias/SampleKmsKey"
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "AWS Internal",
      "type": "AWS::KMS::Key",
    }
  ]
}
```

```
"ARN": "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

進一步了解

下列資源提供有關靜態資料加密的詳細資訊：

- 如需有關AWS KMS概念的詳細資訊，請參閱AWS Key Management Service開發人員指南中的[AWS Key Management Service基本概念](#)。
- 如需有關AWS KMS安全性的詳細資訊，請參閱AWS Key Management Service開發人員指南AWS Key Management Service中的[的安全性最佳做法](#)。

彈性叢集中的服務連結角色

Amazon DocumentDB 彈性叢集使用 AWS Identity and Access Management (IAM) [服務](#)連結角色。服務連結角色是一種獨特的 IAM 角色類型，可直接連結至 Amazon DocumentDB 彈性叢集。服務連結角色由 Amazon DocumentDB 彈性叢集預先定義，並包含服務代表您呼叫其他AWS服務所需的所有許可。

服務連結角色可讓使用 Amazon DocumentDB 彈性叢集變得更加輕鬆，因為您不必手動新增必要的許可。Amazon DocumentDB 彈性叢集會定義其服務連結角色的許可，除非另有定義，否則只有 Amazon DocumentDB 彈性叢集可以擔任其角色。定義的許可包括信任政策和許可政策，並且該許可政策不能附加到任何其他 IAM 實體。您必須先刪除角色的相關資源，才能刪除角色。這可以保護您的 Amazon DocumentDB 彈性叢集資源，因為您無法不小心移除存取資源的權限。

如需支援服務連結角色之其他服[AWS務的相關資訊](#)，請參閱[使用 IAM](#) 的服務，並在服務連結角色欄中尋找標示為是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

彈性叢集的服務連結角色權限

Amazon DocumentDB 彈性叢集使用名為的服務連結角色，AWSServiceRoleForDocDB-Elastic允許 Amazon DocumentDB 彈性叢集代表叢集呼叫AWS服務。

此服務連結角色具有名為 AmazonDocDB-ElasticServiceRolePolicy 的許可政策，該政策會授予此角色在帳戶中操作的許可。角色許可政策允許 Amazon DocumentDB 彈性叢集在指定的資源上完成下列動作：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": [
            "AWS/DocDB-Elastic"
          ]
        }
      }
    }
  ]
}
```

Note

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如果您遇到下列錯誤訊息：「無法建立資源。請確認您擁有建立服務連結角色的許可。否則請等待，然後稍後再試。」，請確定您已啟用下列權限：

```
{
  "Action": "iam:CreateServiceLinkedRole",
  "Effect": "Allow",
  "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
  "Condition": {
    "StringLike": {
```



```
"iam:AWSServiceName":"docdb-elastic.amazonaws.com"  
  }  
}  
}
```

如需詳細資訊，請參閱 [AWS Identity and Access Management 使用者指南](#) 中的 [服務連結角色權限](#)。

為 Amazon DocumentDB 彈性叢集建立服務連結角色

您不需要手動建立一個服務連結角色。當您建立資料庫執行個體時，Amazon DocumentDB 彈性叢集會為您建立服務連結角色。

編輯 Amazon DocumentDB 彈性叢集的服務連結角色

Amazon DocumentDB 彈性叢集不允許您編輯 `AWSServiceRoleForDocDB-Elastic` 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱 [AWS Identity and Access Management 使用者指南](#) 中的 [編輯服務連結角色](#)。

刪除 Amazon DocumentDB 彈性叢集的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。不過您必須先刪除您的所有叢集，才能刪除服務連結角色。

清除服務連結角色

您必須先確認服務連結角色沒有作用中的工作階段，並移除該角色使用的資源，之後才能使用 IAM 將其刪除。

若要檢查服務連結角色在 IAM 主控台中是否具有作用中工作階段：

1. 登入 [AWS Management Console](#) 並開啟 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)。然後選擇 `AWSServiceRoleForDocDB-Elastic` 角色的名稱 (而非核取方塊)。
3. 在所選角色的 Summary (摘要) 頁面中，選擇 Access Advisor (存取 Advisor) 分頁。

Note

如果您不確定 Amazon DocumentDB 彈性叢集是否正在使用該 `AWSServiceRoleForDocDB-Elastic` 角色，可以嘗試刪除該角色。如果服務正在使用角色，則刪除會失敗，您可 AWS 區

域以檢視角色的使用位置。如果服務正在使用該角色，您必須先等到工作階段結束，才能刪除該角色。您無法撤銷服務連結角色的工作階段。

如果要移除AWSServiceRoleForDocDB-Elastic角色，必須先刪除所有叢集。

刪除您的所有叢集

若要刪除 Amazon DocumentDB 主控台中的叢集：

1. 登入[AWS Management Console](#)並開啟 Amazon DocumentDB 主控台。
2. 在導覽窗格中，選擇叢集。
3. 選擇您要刪除的叢集。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 如果系統提示您建立最終快照？，選擇「是」或「否」。
6. 如果您在前一個步驟中選擇 Yes (是)，則對於 Final snapshot name (最終快照名稱)，輸入您的最終快照名稱。
7. 選擇刪除。

Note

您可以使用 IAM 主控台、IAM CLI 或 IAM API 刪除 AWSServiceRoleForDocDB-Elastic 服務連結角色。如需詳細資訊，請參閱《AWS Identity and Access Management 使用者指南》中的[刪除服務連結角色](#)。

Amazon DocumentDB

監控您的AWS服務是保持系統健康和最佳運作的重要組成部分。明智的收集解AWS解決方案各方面的監控資料，以便在失敗或降級情況下，更輕鬆地偵錯和修正失敗或降級。在開始監視AWS解決方案之前，我們建議您考慮並制定下列問題的答案：

- 監控目標是什麼？
- 您要監控哪些資源？
- 監控這些資源的頻率為何？
- 要使用哪些監控工具？
- 誰負責進行監控？
- 若發生錯誤，誰會收到通知以及透過何種方式？

若要了解目前的效能模式、識別效能異常，並制定方法來解決問題，您應該針對各種不同的時間、依不同的負載條件建立基準效能指標。在監控AWS解決方案時，建議您儲存歷史監視資料，以供 future 參考和建立基準。

一般來說，效能指標的可接受值視您的基準看起來像什麼，以及您的應用程式正在做什麼而定。調查距離基準的一致或趨勢變異。以下是關於特定類型指標的建議：

- 高 CPU 或 RAM 消耗量 — CPU 或 RAM 使用量的高值若是符合應用程式的目標 (如輸送量或並行)，而且是預期的，這樣的消耗量就可能是合理的。
- 儲存磁碟區消耗量 — 如果使用的空間持續保持在等於或高於總儲存磁碟區空間的 85%，請調查儲存消耗量 (VolumeBytesUsed)。判斷您是否能從儲存磁碟區刪除資料，或將資料存檔至不同的系統來釋出空間。如需詳細資訊，請參閱 [Amazon DocumentDB 存儲](#) 及 [Amazon DocumentDB 配額和限制](#)。
- 網路流量 — 對於網路流量，請洽系統管理員，以了解您的網域網路和網際網路連線預期的輸送量。調查網路流量的傳輸量是否如預期一致地降低。
- 資料庫連線 — 如果您看到大量使用者連線，同時執行個體效能下降且回應時間延長，請考慮限制資料庫連線。資料庫執行個體使用者連接的最佳數量，將因執行個體類別和要執行的操作複雜性而不同。
- IOPS 指標 — IOPS 指標的預期值視磁碟規格和伺服器組態而定，因此請使用您的基準來了解何謂典型。調查值是否與您的基準一致地不同。為獲得最佳 IOPS 效能，請確定您的一般工作集將放入記憶體中，以將讀取和寫入操作降到最低。

Amazon DocumentDB CloudWatch Amazon DocumentDB 您可以使用各種工具來檢視 Amazon 文件資料庫指標，包括 Amazon DocumentDB 主控台 AWS CLI、CloudWatch API 和 Performance Insights。

主題

- [監控 Amazon DocumentDB 集群的狀態](#)
- [監控 Amazon DocumentDB 執行個體的狀態](#)
- [檢視 Amazon DocumentDB](#)
- [使用 Amazon DocumentDB 事件訂閱](#)
- [使用監控亞馬遜文檔數據庫 CloudWatch](#)
- [使用記錄 Amazon DocumentDB API 呼叫 AWS CloudTrail](#)
- [分析亞馬遜文檔數據庫操作](#)
- [使用績效詳情進行監控](#)

監控 Amazon DocumentDB 集群的狀態

根據叢集運作狀態的叢集狀態。您可以使用 Amazon DocumentDB AWS CLI `describe-db-clusters`

主題

- [叢集狀態值](#)
- [監視叢集的狀態](#)

叢集狀態值

下表列出叢集狀態的有效值。

| 叢集狀態 | 描述 |
|-----------|-----------------------------|
| active | 叢集處於作用中狀態。此狀態僅適用於彈性叢集。 |
| available | 叢集運作正常且可供使用。此狀態僅套用到執行個體型叢集。 |

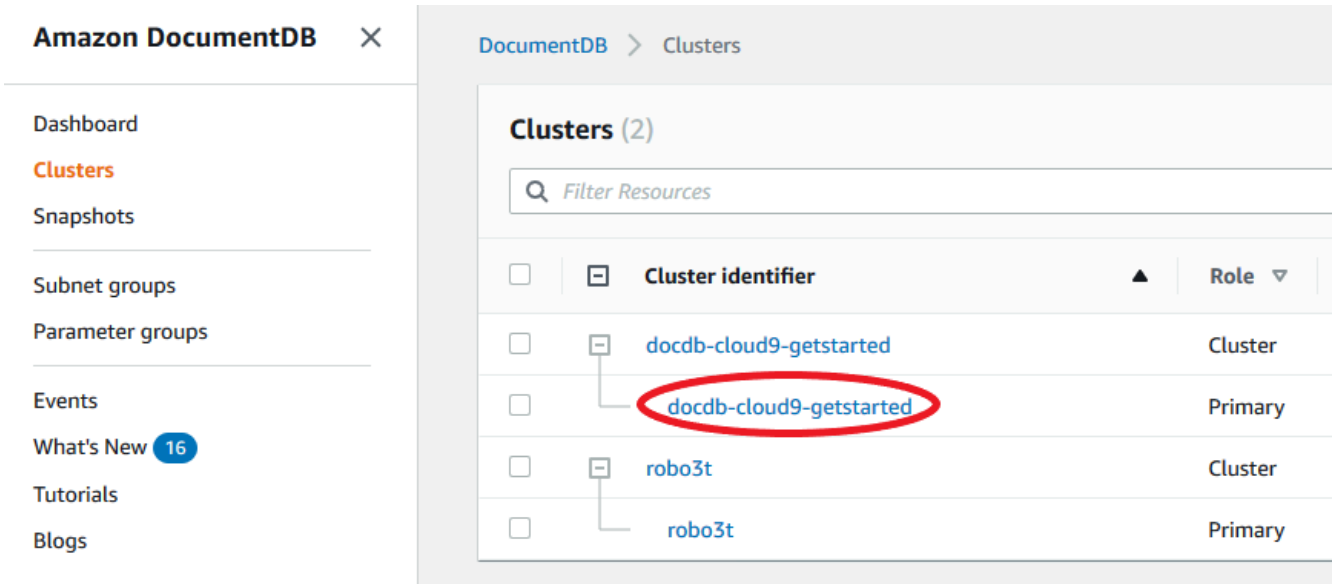
| 叢集狀態 | 描述 |
|-------------------------------------|--------------------------------------|
| backing-up | 目前正在備份資料庫叢集。 |
| creating | 正在建立叢集。正在建立時無法存取它。 |
| deleting | 正在刪除叢集。正在建立時無法將它刪除。 |
| failing-over | 正在從主要執行個體容錯移轉至 Amazon DocumentDB |
| inaccessible-encryption-credentials | 無法存取用來加密或解密叢集的 AWS KMS 金鑰。 |
| maintenance | 正在將維護更新套用到叢集。此狀態用於 Amazon DocumentDB |
| migrating | 正在將叢集快照還原至叢集。 |
| migration-failed | 無法進行遷移。 |
| modifying | 客戶請求修改叢集，因此正在進行修改。 |
| renaming | 客戶請求重新命名叢集，因此正在重新命名。 |
| resetting-master-credentials | 客戶請求重新設定叢集的主要登入資料，因此正在重新設定。 |
| upgrading | 正在升級叢集引擎版本。 |

監視叢集的狀態

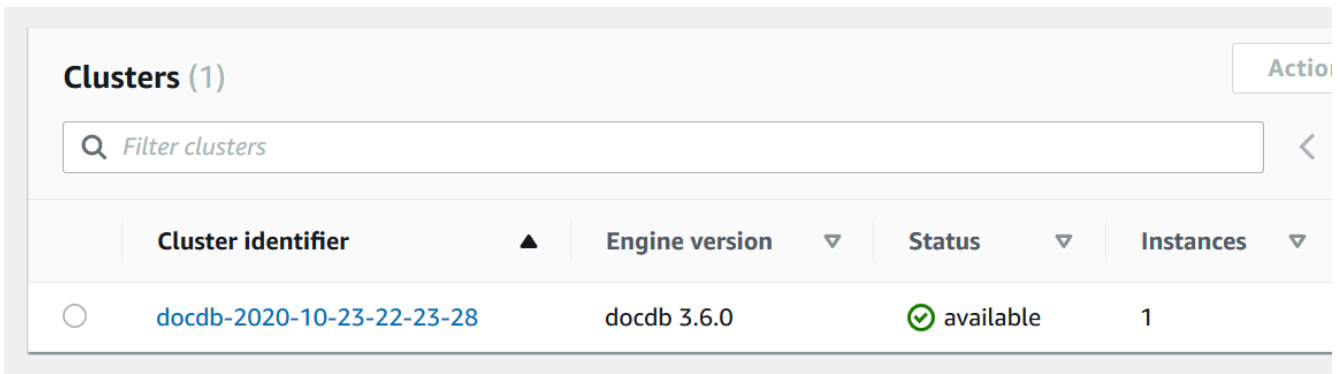
Using the AWS Management Console

使用 AWS Management Console 判斷叢集的狀態時，請使用下列程序。

1. 登入AWS Management Console，開啟位於 <https://console.aws.amazon.com/docdb> Amazon DocumentDB
2. 在導覽窗格中，選擇 Clusters (叢集)。
3. 在 [叢集] 導覽方塊中，您會看到 [叢集識別碼] 資料行。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。



4. 在 [叢集識別碼] 資料行中，尋找您感興趣的執行個體名稱。然後，若要尋找執行個體的状态，請將該列讀取至「状态」欄，如下所示。



Using the AWS CLI

使用 AWS CLI 判斷叢集的状态時，請使用 `describe-db-clusters` 操作。以下程式碼會尋找叢集 `sample-cluster` 的状态。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \
```

```
--db-cluster-identifier sample-cluster \  
--query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

針對 Windows :

```
aws docdb describe-db-clusters ^  
--db-cluster-identifier sample-cluster ^  
--query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

此操作的輸出將會如下所示。

```
[  
  [  
    "sample-cluster",  
    "available"  
  ]  
]
```

監控 Amazon DocumentDB 執行個體的狀態

Amazon DocumentDB 提供資料庫中每個已設定執行個體目前狀況的相關資訊。

您可以檢視 Amazon DocumentDB 執行個體的三種狀態類型：

- **執行處理狀態**：此狀態會顯示在叢集表格的「狀態」資料欄中，AWS Management Console 並顯示執行處理目前的生命週期狀況。「狀態」欄中顯示的值衍生自 DescribeDBCluster API 回應的 Status 欄位。
- **執行處理狀況狀態**：此狀態會顯示在叢集表格的「執行處理狀況」資料欄中，AWS Management Console 並顯示資料庫引擎 (負責管理和擷取資料的元件) 是否在執行中。執行個體運作狀態欄中顯示的值是以 AmazonCloudWatchEngineUptime 系統指標為基礎。
- **維護狀態**：此狀態會顯示在叢集表格的「維護」資料欄中，AWS Management Console 並指出需要套用至執行處理之任何維護事件的狀態。維護狀態與其他執行個體狀態無關，且衍生自 PendingMaintenanceAction API。如需維護狀態的詳細資訊，請參閱 [維護 Amazon DocumentDB](#)。

主題

- [執行個體狀態值](#)

- [使用AWS Management Console或監視執行個體狀態AWS CLI](#)
- [執行個體運作狀態值](#)
- [監視執行個體健全狀態AWS Management Console](#)

執行個體狀態值

下表列出執行個體的可能狀態值，以及每種狀態的計費方式。表中列出是否將依執行個體和儲存體計費、只針對儲存體計費、或不計費。在所有執行個體狀態下，一律要為備份用途付費。

| 執行個體 | 計費 | 描述 |
|-------------------------------------|-----|--|
| available | 計費 | 執行個體運作正常可供使用。 |
| backing-up | 計費 | 目前正在備份執行個體。 |
| configuring-log-exports | 計費 | 針對此執行個體，正在啟用或停用將 CloudWatch 日誌檔案發佈至 Amazon Locugs。 |
| creating | 不計費 | 正在建立執行個體。當執行個體正在建立時，無法存取執行個體。 |
| deleting | 不計費 | 正在刪除執行個體。 |
| failed | 不計費 | 執行個體失效，Amazon DocumentDB 若要復原資料，請執行point-in-time還原到執行個體的最後可還原時間。 |
| inaccessible-encryption-credentials | 不計費 | 無法存取用來加密或解密執行個體的 AWS KMS 金鑰。 |
| incompatible-network | 不計費 | Amazon DocumentDB 例如，如果子網路中的所有可用 IP 地址都在使用中，Amazon DocumentDB |

| 執行個體 | 計費 | 描述 |
|--------------------------------------|--------|--|
| maintenance | 計費 | Amazon DocumentDB 此狀態用於 Amazon DocumentDB 我們正在評估透過此狀態對客戶公開其他維護動作的方式。 |
| modifying | 計費 | 由於請求修改執行個體，因此正在修改執行個體。 |
| rebooting | 計費 | 由於需要重新啟動執行個體，正在重新 Amazon DocumentDB 行個體，正在重新啟動執行個體。 |
| renaming | 計費 | 由於請求重新命名執行個體，因此正在重新命名執行個體。 |
| resetting -master-c redentials | 計費 | 由於請求重設執行個體的主要登入資料，因此正在重設主要登入資料。 |
| restore-error | 計費 | 執行個體在嘗試還原到point-in-time或從快照還原時發生錯誤。 |
| starting | 針對儲存計費 | 執行個體正在啟動。 |
| stopped | 針對儲存計費 | 執行個體已停止。 |
| stopping | 針對儲存計費 | 正在停止執行個體。 |
| storage-full | 計費 | 執行個體已達到其配置的儲存容量。這是一個重要的狀態，且應立即解決；請修改執行個體以擴展您的儲存。設定 AmazonCloudWatch 警示在儲存空間變少時警告您，以免遇到此情況。 |

使用AWS Management Console或監視執行個體狀態AWS CLI

使用AWS Management Console或AWS CLI來監視執行個體的狀態。

Using the AWS Management Console

使用 AWS Management Console 判斷叢集的狀態時，請使用下列程序。

1. 登入AWS Management Console，開啟位於 <https://console.aws.amazon.com/docdb> Amazon DocumentDB
2. 在導覽窗格中，選擇 Clusters (叢集)。

Note

請注意，在 [叢集] 導覽方塊中，資料行叢集識別碼會同時顯示叢集和執行個體。執行個體會列在叢集下方，與下圖類似。

| Cluster identifier | Role | Engine version | Region & AZ |
|-------------------------|---------|----------------|-------------|
| docdb-cloud9-getstarted | Cluster | 3.6.0 | us-east-1 |
| docdb-cloud9-getstarted | Primary | 3.6.0 | us-east-1f |
| robo3t | Cluster | 3.6.0 | us-east-1 |
| robo3t | Primary | 3.6.0 | us-east-1d |

3. 尋找您感興趣的執行個體名稱。然後，閱讀該列上的 Status (狀態) 欄找出執行個體的状态，如下所示。

| Cluster identifier | Role | Engine version | Region & AZ | Status |
|-------------------------|---------|----------------|-------------|-----------|
| docdb-cloud9-getstarted | Cluster | 3.6.0 | us-east-1 | available |
| docdb-cloud9-getstarted | Primary | 3.6.0 | us-east-1f | available |
| robo3t | Cluster | 3.6.0 | us-east-1 | available |
| robo3t | Primary | 3.6.0 | us-east-1d | available |

Using the AWS CLI

使用 AWS CLI 判斷叢集的狀態時，請使用 `describe-db-instances` 操作。以下程式碼會尋找執行個體 `sample-cluster-instance-01` 的狀態。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \
  --db-instance-identifier sample-cluster-instance-01 \
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

針對 Windows：

```
aws docdb describe-db-instances ^
  --db-instance-identifier sample-cluster-instance-01 ^
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

此操作的輸出將會如下所示。

```
[
  [
    "sample-cluster-instance-01",
    "available"
  ]
]
```

執行個體運作狀態值

下表列出執行個體的可能運作狀態值。「執行處理健全狀況」資料欄位於的「叢集」表格中 AWS Management Console，會顯示資料庫引擎 (負責儲存、管理和擷取資料的元件) 是否正常運作。此資料欄也指出中 CloudWatch 可用的 EngineUptime 系統測量結果是否顯示每個執行處理的狀況狀態。

| 執行個體運作狀態 | 描述 |
|----------|------------------------------------|
| 狀態良好 | 資料庫引擎正在 Amazon DocumentDB 執行個體中執行。 |
| 不健康 | 資料庫引擎未執行，或在不到一分鐘前重新啟動。 |

監視執行個體健全狀態AWS Management Console

使用AWS Management Console來監視執行個體的健全狀況狀態。

使用時AWS Management Console，請遵循下列步驟瞭解執行個體的健全狀態。

1. 登入AWS Management Console，開啟位於 <https://console.aws.amazon.com/docdb> Amazon DocumentDB
2. 在導覽窗格中，選擇 Clusters (叢集)。

Note

在 [叢集] 導覽方塊中，資料行叢集識別碼會同時顯示叢集和執行個體。執行個體會列在叢集下方，與下圖類似。

| Cluster identifier | Role | Engine version | Region & AZ |
|-------------------------|---------|----------------|-------------|
| docdb-cloud9-getstarted | Cluster | 3.6.0 | us-east-1 |
| docdb-cloud9-getstarted | Primary | 3.6.0 | us-east-1f |
| robo3t | Cluster | 3.6.0 | us-east-1 |
| robo3t | Primary | 3.6.0 | us-east-1d |

3. 尋找您感興趣的執行個體名稱。然後，若要尋找執行個體的狀態，請將該列讀取至「執行個體健全狀況」欄，如下圖所示：

Clusters (4) 🔄

🔍 Filter Resources

| <input type="checkbox"/> | Cluster identifier ▲ | Role ▼ | Engine version ▼ | Region & AZ ▼ | Status ▼ | Instance health | CPU |
|--------------------------|----------------------------|-------------------|------------------|---------------|-------------|-----------------|---------|
| <input type="checkbox"/> | iad-fra-global-cluster | Global cluster | 4.0.0 | 2 regions | 🟢 available | - | - |
| <input type="checkbox"/> | docdb-2023-03-27-11-56-04 | Primary cluster | 4.0.0 | us-east-1 | 🟢 available | - | - |
| <input type="checkbox"/> | docdb-2023-03-27-11-56-04 | Primary instance | 4.0.0 | us-east-1a | 🟢 available | 🟢 healthy | 📊 5.58% |
| <input type="checkbox"/> | docdb-2023-03-27-11-56-042 | Replica instance | 4.0.0 | us-east-1d | 🟢 available | 🟢 healthy | 📊 5.79% |
| <input type="checkbox"/> | docdb-2023-03-27-11-56-043 | Replica instance | 4.0.0 | us-east-1b | 🟢 available | 🟢 healthy | 📊 5.68% |
| <input type="checkbox"/> | docdb-2023-03-27-12-02-55 | Secondary cluster | 4.0.0 | eu-central-1 | 🟢 available | - | - |
| <input type="checkbox"/> | docdb-2023-03-27-12-02-55 | Replica instance | 4.0.0 | eu-central-1c | 🟢 available | 🟢 healthy | 📊 5.88% |
| <input type="checkbox"/> | docdb-2023-03-27-12-02-552 | Replica instance | 4.0.0 | eu-central-1a | 🟢 available | 🟢 healthy | 📊 5.97% |
| <input type="checkbox"/> | docdb-2023-03-28-09-45-05 | Regional cluster | 5.0.0 | us-east-1 | ⏸ stopped | - | - |
| <input type="checkbox"/> | docdb-2023-03-28-09-45-05 | Replica instance | 5.0.0 | us-east-1d | ⏸ stopped | 🔴 unhealthy | - |
| <input type="checkbox"/> | docdb-2023-03-28-09-45-052 | Replica instance | 5.0.0 | us-east-1a | ⏸ stopped | 🔴 unhealthy | - |
| <input type="checkbox"/> | docdb-2023-03-28-09-45-053 | Primary instance | 5.0.0 | us-east-1b | ⏸ stopped | 🔴 unhealthy | - |

📌 Note

執行個體健康狀態輪詢每 60 秒進行一次，並根據 CloudWatch EngineUptime 系統指標進行。「執行個體健全狀況」欄中的值會自動更新。

檢視 Amazon DocumentDB

Amazon DocumentDB 這些建議事項會分析叢集和執行個體組態，以提供最佳實務指南。

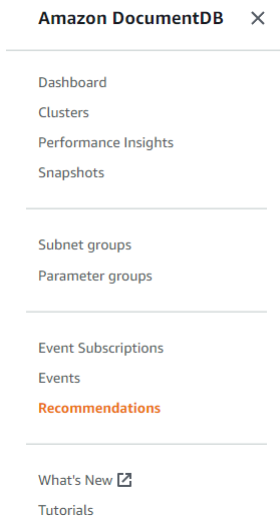
這些建議範例，請參閱下列主題：

| 類型 | 描述 | 建議 | 其他資訊 |
|---------|-------------|---|--|
| 一個 執行個體 | 叢集只包含一個執行個體 | 效能和可用性：我們建議新增另一個執行個體具有相同的執行個體，並位於不同的可用區域。 | Amazon DocumentDB 的高可用性和複寫 |

Amazon DocumentDB Amazon DocumentDB

若要檢視和對 Amazon DocumentDB 建議採取行動

1. 登入AWS Management Console，開啟位於 <https://console.aws.amazon.com/docdb> Amazon DocumentDB
2. 在導覽窗格中，選擇建議：



3. 在「建議」對話方塊中，展開感興趣的區段，然後選取建議的工作。

在以下範例中，建議的任務適用於只有一個執行個體的 Amazon DocumentDB 叢集。建議您新增另一個執行個體以改善效能和可用性。

Recommendations

Recommendations - (1)


▼ DocumentDB Clusters with only one DB Instance (1)

DocumentDB clusters that only have one DB instance. Use more than one DB instance for improved performance and availability.

Clusters

[Apply now](#)

< 1 > 

| Resource Identifier | Recommendation |
|---|--|
|  docdb-2022-01-18-16-55-31 | Add another DB Instance with instance class db.t4g.medium to |

4. 按一下立即套用。

在此範例中，會顯示 [新增實例] 對話方塊：

DocumentDB > Clusters > Add Instances

Add instances to: docdb-2022-01-18-16-55-31

Instance settings

You can create up to 16 instances for a cluster (one primary and 15 replicas).
'docdb-2022-01-18-16-55-31' cluster currently has 1/16 instances.

| | | | |
|--|---|--|---------------------------------------|
| Instance identifier Info | Instance class Info | Promotion tier Info | |
| <input type="text" value="docdb-2022-01-18-16-5"/> | <input type="text" value="db.t3.medium (fre..."/> | <input type="text" value="No preference"/> | <input type="button" value="Remove"/> |

Specify a unique instance identifier.

You can create 14 more instances.

5. 修改新執行個體的設定，然後按一下 [建立]。

使用 Amazon DocumentDB 事件訂閱

Amazon DocumentDB 使用 Amazon Simple Notification Service (Amazon SNS) 在 Amazon DocumentDB 事件發生時提供通知。這些通知可以採用任何 Amazon SNS 在支援的形式 AWS 區域，例如電子郵件、文字訊或呼叫 HTTP 端點。

Amazon DocumentDB 將這些事件分成幾個類別供您訂閱，在該類別的事件發生時，您將會收到通知。您可以訂閱執行個體、叢集、快照、叢集快照或參數群組的事件類別。例如，如果您訂閱指定的執行個體的 Backup 類別，當發生會影響該資料庫執行個體的備份相關事件時，您將會收到通知。當事件訂閱變更時，您也會收到通知。

事件發生於叢集和執行個體層級，因此，您只要訂閱叢集或一個個體，就會收到事件。

事件訂閱會傳送到您在建立訂閱時提供的地址。您可能想要建立多個不同的訂閱，例如接收所有事件通知，而另一個訂閱僅包含您的生產資料庫執行個體的重要事件。您可以輕鬆關閉通知，而無需刪除訂閱。若要這樣做，請在 Amazon DocumentDB 主控台中將 [已啟用] 選項按鈕設定為 [否]。

⚠ Important

Amazon DocumentDB 不保證事件串流中傳送事件的順序。事件順序可能會改變。

Amazon DocumentDB 使用 Amazon SNS 主題的 Amazon Resource Name (ARN) 來識別每個訂閱。Amazon DocumentDB 主控台會在您建立訂閱時為您建立 ARN。

Amazon DocumentDB 事件訂閱的帳單是透過 Amazon SNS 傳送。使用事件通知需要支付 Amazon SNS 費用。如需詳細資訊，請參閱 Amazon Simple Notification Service 定價。除了 Amazon SNS 費用之外，Amazon DocumentDB 不收取事件訂閱費用。

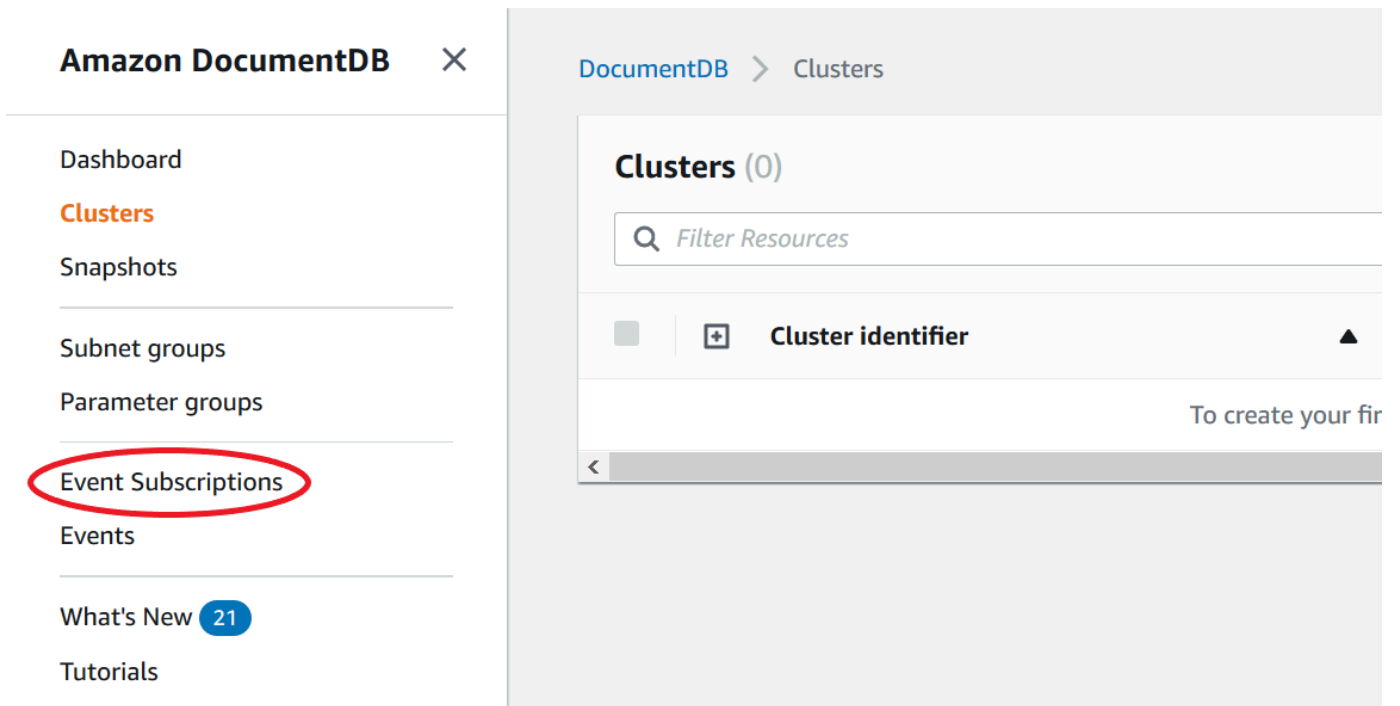
主題

- [訂閱 Amazon 文件資料庫事件訂閱](#)
- [管理 Amazon DocumentDB 事件通知訂閱](#)
- [Amazon DocumentDB 事件類別和消息](#)

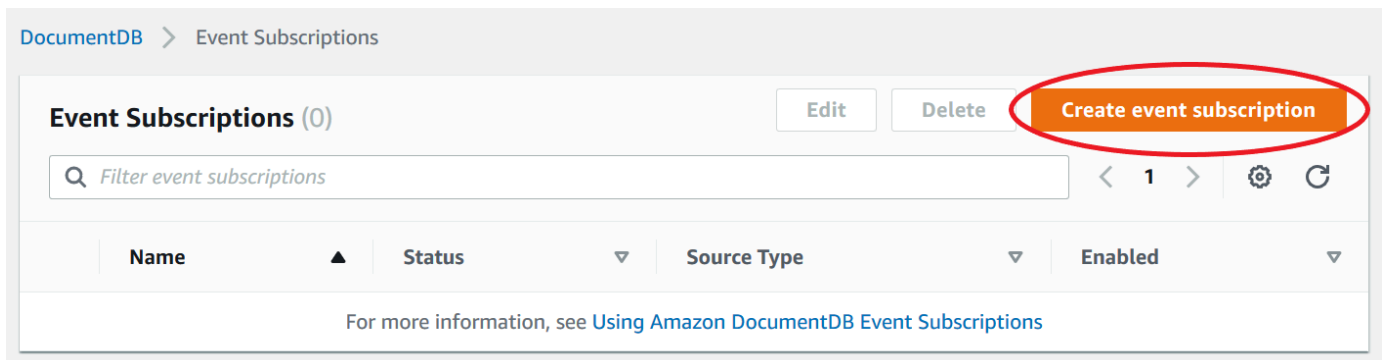
訂閱 Amazon 文件資料庫事件訂閱

您可以使用 Amazon DocumentDB 主控台訂閱事件訂閱，如下所示：

1. 前往 <https://console.aws.amazon.com/docdb> 登入 AWS Management Console。
2. 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。



3. 在 Event subscriptions (事件訂閱) 窗格中，選擇 Create event subscription (建立事件訂閱)。



4. 在 Create event subscription (建立事件訂閱) 對話方塊中，執行下列動作：
 - 在 Name (名稱) 中，輸入事件通知訂閱的名稱。

DocumentDB > Event Subscriptions > Create event subscription

Create event subscription

Details

Name

Name of the subscription

Test

- 在 Target 中，選擇您要傳送通知的目標位置。您可以選擇現有的 ARN 或選擇 [新增電子郵件主題] 來輸入主題名稱和收件者清單。

Target

Send notifications to

ARN

New Email Topic

ARN

ARN to send notifications to

Choose ARN

- 對來源選擇來源類型。根據您選擇的來源類型，選擇您要接收事件通知的事件類別和來源。

Source

Source Type

Source type of resource this subscription will consume events from

Choose source type

- 選擇 建立。

Source

Source Type
Source type of resource this subscription will consume events from

Instances ▼

Instances to include
Instances that this subscription will consume events from

All instances
 Select specific instances

Event Categories to include
Event Categories that this subscription will consume events from

All event categories
 Select specific event categories

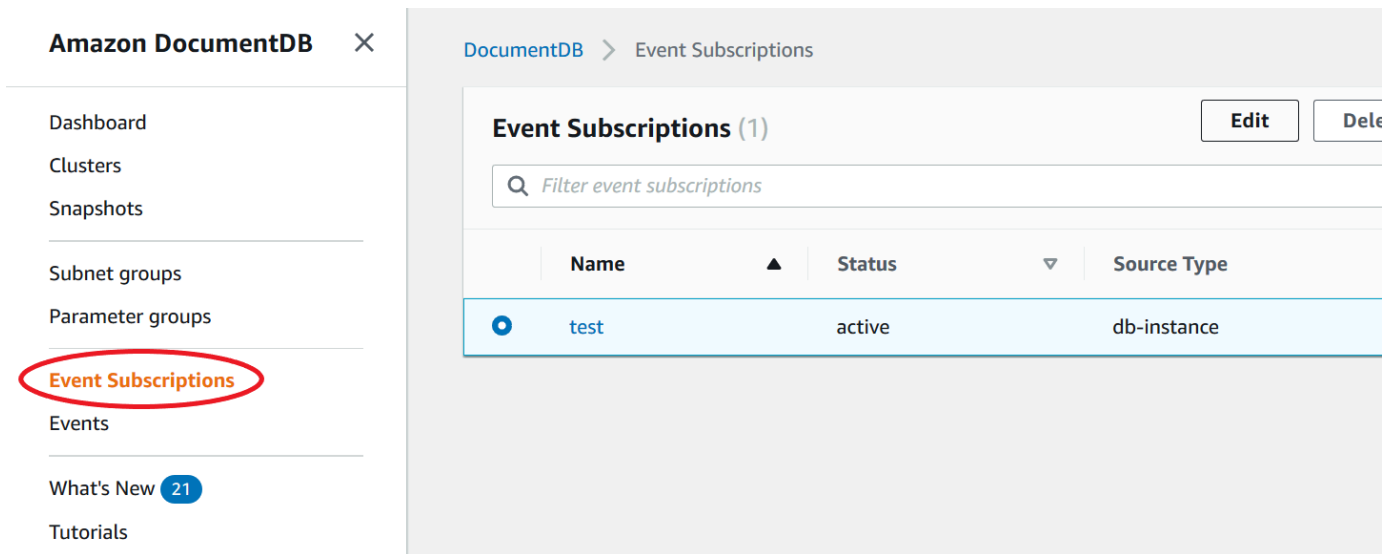
Cancel **Create**

管理 Amazon DocumentDB 事件通知訂閱

如果您在 Amazon DocumentDB 主控台的導覽窗格中選擇事件訂閱，則可以檢視訂閱類別和目前訂閱的清單。您也可以修改或刪除特定訂閱。

修改您目前的 Amazon DocumentDB 事件通知訂閱

1. 前往 <https://console.aws.amazon.com/docdb> 登入 AWS Management Console。
2. 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。Event subscriptions (事件訂閱) 窗格顯示所有事件通知訂閱。



Amazon DocumentDB

- Dashboard
- Clusters
- Snapshots
- Subnet groups
- Parameter groups
- Event Subscriptions**
- Events
- What's New **21**
- Tutorials

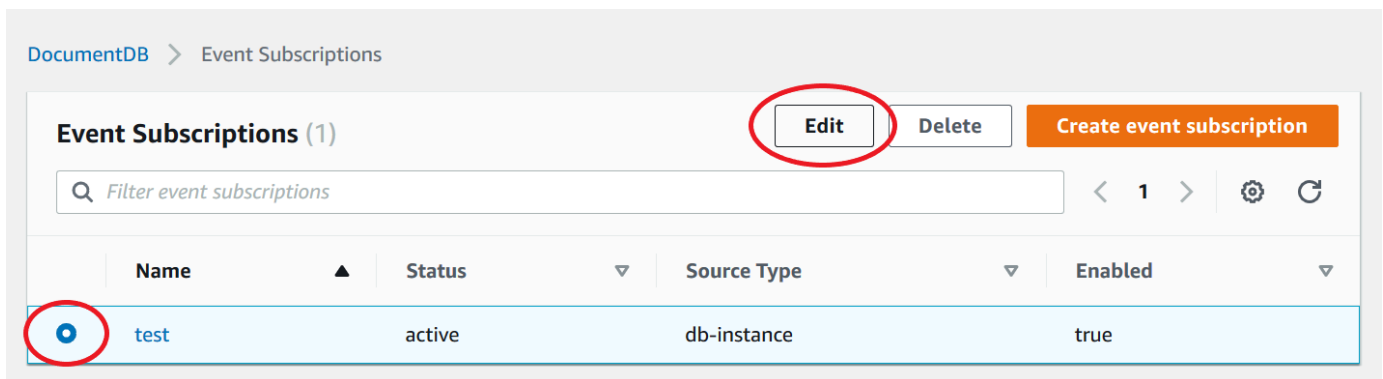
DocumentDB > Event Subscriptions

Event Subscriptions (1) Edit Delete

Filter event subscriptions

| Name | Status | Source Type |
|------|--------|-------------|
| test | active | db-instance |

- 在 Event subscriptions (事件訂閱) 窗格中，選擇您要修改的訂閱，然後選擇 Edit (編輯)。



DocumentDB > Event Subscriptions

Event Subscriptions (1) Edit Delete Create event subscription

Filter event subscriptions

| Name | Status | Source Type | Enabled |
|------|--------|-------------|---------|
| test | active | db-instance | true |

- 在 Target (目標) 或 Source (來源) 區段中，對訂閱進行變更。您可以在來源區段中選取或取消選取來源識別碼，以新增或移除來源識別碼。

Modify event subscription

Details

Enabled

- Enabled
 Disabled

Target

Send notifications to

- ARN
 New Email Topic

ARN

ARN to send notifications to

Test

5. 選擇 Modify (修改)。Amazon DocumentDB 主控台顯示正在修改訂閱。

Event Categories to include

Event Categories that this subscription will consume events from

- All event categories
 Select specific event categories

Cancel

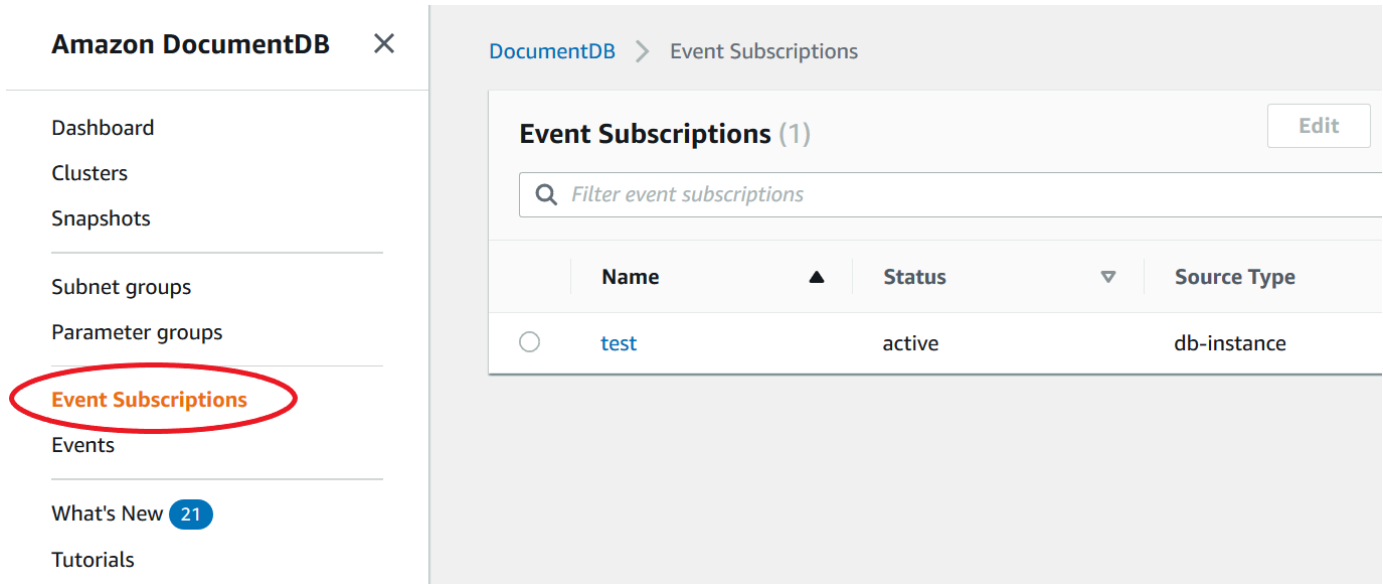
Modify

刪除 Amazon 文件資料庫事件通知訂閱

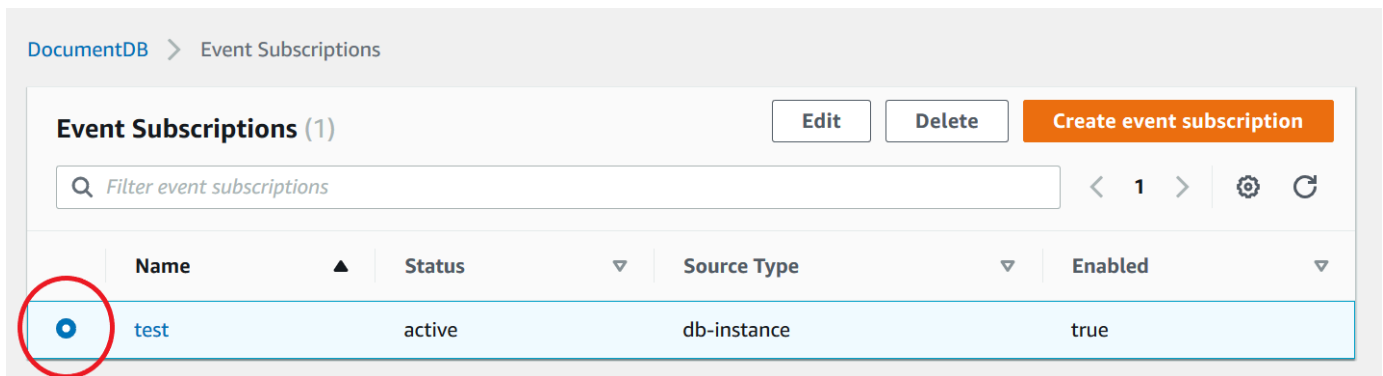
您可以刪除不再需要的訂閱。該主題的所有訂閱者將不會再收到該訂閱指定的事件通知。

1. 前往 <https://console.aws.amazon.com/doccdb> 登入 AWS Management Console。

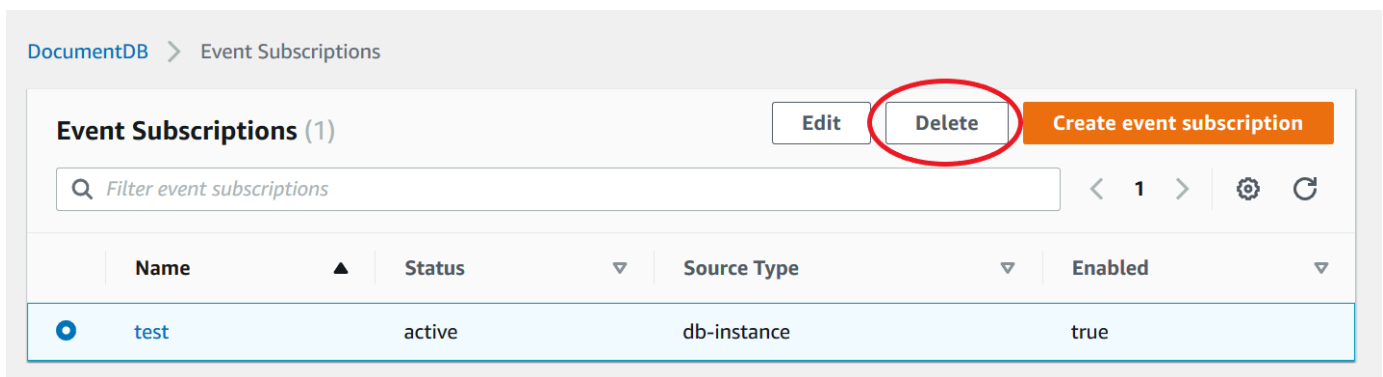
- 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。



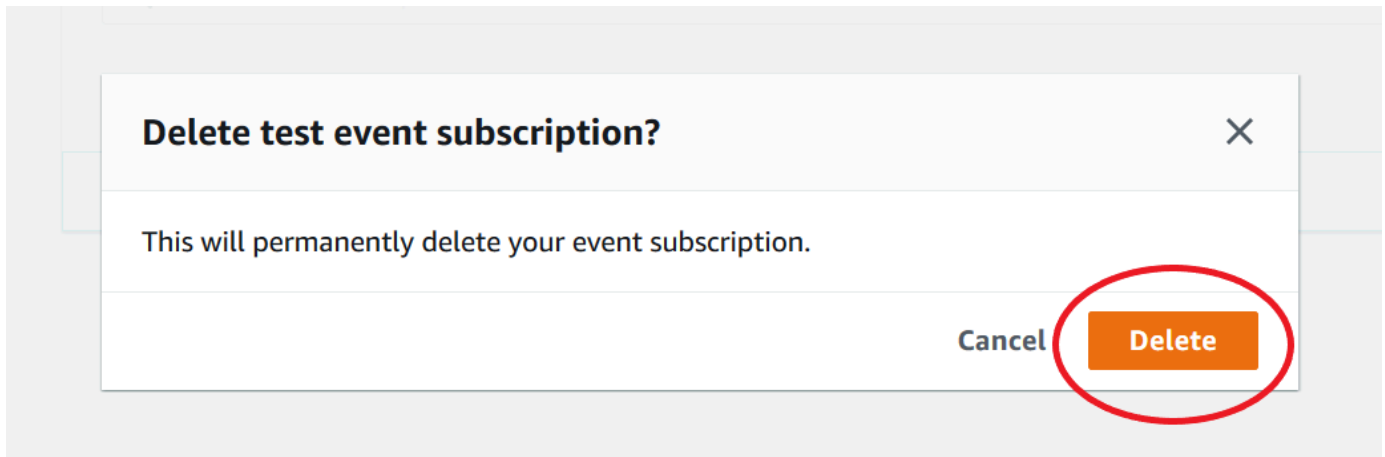
- 在事件訂閱窗格中，選擇您要刪除的訂閱。



- 選擇 刪除。



- 將出現一個彈出窗口，詢問您是否要永久刪除此通知。選擇 刪除。



Amazon DocumentDB 事件類別和消息

Amazon DocumentDB 會在各類別中產生大量的事件，您可以使用主控台訂閱這些類別。每個品類都適用於來源類型，其可以是例證、快照或參數群組。

Note

Amazon DocumentDB 使用現有的亞馬遜 RDS 事件定義和 ID。

源自執行個體的 Amazon DocumentDB 事件

| 類別 | 描述 |
|------|----------------|
| 可用性 | 實例已重新啟動。 |
| 可用性 | 執行個體關機。 |
| 設定更改 | 套用修改至實體類別。 |
| 設定更改 | 已完成將修改套用至實體類別。 |
| 設定更改 | 重設主要證明資料。 |
| 建立 | 已建立例證。 |
| 刪除 | 已刪除實例 |

| 類別 | 描述 |
|---------|---|
| 失敗 | 因為不相容的組態或基礎儲存問題，執行個體已失敗。為執行個 point-in-time-restore 體開始執行個體。 |
| 則通知 | 實例已停止。 |
| 則通知 | 實例已啟動。 |
| 則通知 | 因為超過允許停止的時間上限，正在啟動。 |
| 復原 | 執行個體的復原已啟動。復原時間依據恢復的資料量而有不同。 |
| 復原 | 執行個體的復原已完成。 |
| 安全性修補程式 | 作業系統更新適用於您的執行個體中支援。如需套用更新的相關資訊，請參閱 維護 Amazon DocumentDB 。 |

源自叢集的 Amazon DocumentDB 事件

| 類別 | 描述 |
|------|-------------------------|
| 建立 | 叢集已建立 |
| 刪除 | 叢集已刪除。 |
| 容錯移轉 | 再次晉升上一個小學。 |
| 容錯移轉 | 完成容錯移轉至執行個體 |
| 容錯移轉 | 已啟動容錯移轉至資料庫執行個體 |
| 容錯移轉 | 啟動相同 AZ 容錯移轉至資料庫執行個體：%s |
| 容錯移轉 | 啟動跨 AZ 容錯移轉至資料庫執行個體：%s |

| 類別 | 描述 |
|-----|-----------------------|
| 維護 | 叢集已修補。 |
| 維護 | 資料庫叢集處於無法升級的狀態：%s |
| 則通知 | 叢集已停止。 |
| 則通知 | 叢集已啟動。 |
| 則通知 | 叢集停止失敗。 |
| 則通知 | 叢集由於超過允許停止的時間上限，正在啟動。 |
| 則通知 | 將叢集從 %s 重新命名為 %s。 |

源自叢集快照的 Amazon DocumentDB 事件

下表顯示當 Amazon DocumentDB 叢集快照為來源類型時的事件類別和事件清單。

| 類別 | 描述 |
|----|------------|
| 備份 | 建立手動叢集快照。 |
| 備份 | 已建立手動叢集快照。 |
| 備份 | 建立自動叢集快照。 |
| 備份 | 已建立自動叢集快照。 |

Amazon DocumentDB 事件源自參數組

下表顯示當參數群組為來源類型時的事件類別和事件清單。

| 類別 | 描述 |
|------|--------------------------|
| 設定更改 | 已使用套用方法 %s 將參數 %s 更新為 %s |

使用監控亞馬遜文檔數據庫 CloudWatch

亞馬遜文檔數據庫 (與 MongoDB 兼容性) 與亞馬遜集成 CloudWatch 這樣您就可以收集和分析叢集的操作指標。您可以使用 CloudWatch 控制台, 亞馬遜文檔數據庫控制台, AWS Command Line Interface(AWS CLI) , 或 CloudWatchAPI。

CloudWatch 也可讓您設定警示 , 以便在指標值違反您指定的閾值時收到通知。你甚至可以設置亞馬遜 CloudWatch 發生違規時採取糾正措施的事件。有關使用的更多信息 CloudWatch 和鬧鐘 , 請參閱[亞馬遜 CloudWatch 文件](#)。

主題

- [Amazon DocumentDB 指標](#)
- [檢視 CloudWatch 資料](#)
- [亞馬遜文檔數據庫尺寸](#)
- [監視反應](#)
- [監控資料庫連線](#)

Amazon DocumentDB 指標

若要監控 Amazon DocumentDB 叢集和執行個體的運作狀態和效能 , 您可以在 Amazon DocumentDB 主控台中檢視下列指標。

Note

下表中的量度同時適用於執行個體型和彈性叢集。

資源使用率

| 指標 | 描述 |
|----------------------------------|---|
| BackupRetentionPeriodStorageUsed | GiB 中的備份儲存總量用於支援 point-in-time 在亞馬遜文檔數據庫的保留時間內恢復功能。包含在 TotalBackupStorageBilled 指標報告的總數中。針對每個 |

| 指標 | 描述 |
|-------------------------|--|
| | Amazon DocumentDB 叢集單獨計算。 |
| ChangeStreamLogSize | 您的叢集用於存放變更串流日誌的儲存容量 (MB)。此數值是叢集的儲存總量子集 (VolumeBytesUsed)，且會影響叢集費用。如需儲存體定價資訊，請參閱 亞馬遜文檔數據庫產品頁面 。變更串流日誌大小是您叢集所進行之變更程度與變更串流長期保留期間的函數。如需變更串流的詳細資訊，請參閱 使用更改流與亞馬遜文檔數據庫 。 |
| CPUUtilization | 執行個體所使用 CPU 的百分比。 |
| DatabaseConnections | 執行個體上以一分鐘頻率開啟的連線數目。 |
| DatabaseConnectionsMax | 一分鐘期間內，執行處理開啟的資料庫連線數目上限。 |
| DatabaseCursors | 以一分鐘頻率取得的執行個體上開啟的游標數目。 |
| DatabaseCursorsMax | 一分鐘內執行個體開啟游標的數目上限。 |
| DatabaseCursorsTimedOut | 一分鐘內逾時的游標數目。 |
| FreeableMemory | 可用的隨機存取記憶體的数量，以位元組為單位。 |

| 指標 | 描述 |
|------------------------------|---|
| FreeLocalStorage | 此指標回報可供各執行個體用於暫時資料表與記錄的儲存空間量。此值取決於執行個體類別。您可以為執行個體選擇較大的執行個體，以增加執行個體的可用儲存空間量。 |
| LowMemThrottleQueueDepth | 由於以一分鐘頻率佔用的可用記憶體不足而受到限制的要求的佇列深度。 |
| LowMemThrottleMaxQueueDepth | 由於一分鐘內可用記憶體不足而節流之要求的最大佇列深度。 |
| LowMemNumOperationsThrottled | 由於一分鐘內可用記憶體不足而節流的要求數目。 |
| SnapshotStorageUsed | 指定 Amazon DocumentDB 叢集在其備份保留期間以外的所有快照使用的備份儲存體總量 (GiB)。包含在 TotalBackupStorageBilled 指標報告的總數中。針對每個 Amazon DocumentDB 叢集單獨計算。 |
| SwapUsage | 執行個體上使用的交換空間量。 |

| 指標 | 描述 |
|--------------------------|---|
| TotalBackupStorageBilled | GiB 中的備份儲存總量會針對指定的 Amazon DocumentDB 叢集向您收費。包括以 BackupRetentionPeriodStorageUsed 和 SnapshotStorageUsed 指標衡量的備份儲存。針對每個 Amazon DocumentDB 叢集單獨計算。 |
| TransactionsOpen | 以一分鐘頻率執行個體開啟的交易數目。 |
| TransactionsOpenMax | 一分鐘期間內，執行個體開啟的交易數目上限。 |
| VolumeBytesUsed | 您的叢集使用的儲存空間量 (以位元組為單位)。這個值會影響叢集的成本。如需定價資訊，請參閱 亞馬遜文檔數據庫產品頁面 。 |

Latency (延遲)

| 指標 | 描述 |
|----------------------------|--|
| DBClusterReplicaLagMaximum | 主執行個體和叢集中每個 Amazon DocumentDB 執行個體之間的最大延遲量 (以毫秒為單位)。 |
| DBClusterReplicaLagMinimum | 主要執行個體與叢集中每個複寫執行個體之間的最小延遲量 (單位毫秒)。 |

| 指標 | 描述 | |
|----------------------|---------------------------------|--|
| DBInstanceReplicaLag | 從主要執行個體複寫更新至複寫執行個體時的延遲量 (單位毫秒)。 | |
| ReadLatency | 平均每次磁碟輸入/輸出操作耗用的時間量。 | |
| WriteLatency | 平均每次磁碟 I/O 操作耗用的時間量 (單位毫秒)。 | |

操作

| 指標 | 描述 | |
|-------------------|-------------------|--|
| DocumentsDeleted | 一分鐘內已刪除文件的數量。 | |
| DocumentsInserted | 一分鐘期間內插入的文件數目。 | |
| DocumentsReturned | 一分鐘內退回的文件數量。 | |
| DocumentsUpdated | 在一分鐘內更新的文件數目。 | |
| OpcountersCommand | 一分鐘內發出的指令數目。 | |
| OpcountersDelete | 一分鐘內發出的刪除作業數目。 | |
| OpcountersGetmore | 在一分鐘的時間內發出的數量。 | |
| OpcountersInsert | 在一分鐘期間內發出的插入作業數目。 | |
| OpcountersQuery | 在一分鐘期間內發出的查詢數目。 | |

| 指標 | 描述 |
|-----------------------|--------------------------|
| OpcountersUpdate | 一分鐘內發出的更新作業數目。 |
| TransactionsStarted | 一分鐘期間內在執行個體上啟動的交易數目。 |
| TransactionsCommitted | 一分鐘期間內，執行處理確認的交易數目。 |
| TransactionsAborted | 在一分鐘期間內，執行個體中止的交易數目。 |
| TTLDeletedDocuments | TTLMonitor 在一分鐘內刪除的文件數量。 |

輸送量

| 指標 | 描述 |
|---------------------------|--|
| NetworkReceiveThroughput | 叢集中每個執行個體從用戶端接收到的網路傳輸量 (以位元組/秒為單位)。此傳輸量不包含叢集中的執行個體與叢集磁碟區之間的網路流量。 |
| NetworkThroughput | Amazon DocumentDB 叢集中每個執行個體從用戶端接收和傳輸到用戶端的網路輸送量 (以每秒位元組為單位)。此傳輸量不包含叢集中的執行個體與叢集磁碟區之間的網路流量。 |
| NetworkTransmitThroughput | 叢集中每個執行個體傳送至用戶端的網路傳輸量 (以位元組/秒為單位)。此傳輸量不包含叢 |

| 指標 | 描述 |
|----------------|---|
| | 集中的執行個體與叢集磁碟區之間的網路流量。 |
| ReadIOPS | 磁碟讀取輸入/輸出操作的每秒平均次數。亞馬遜文檔數據庫分別報告讀取和寫入 IOPS，每隔一分鐘一次。 |
| ReadThroughput | 平均每秒從磁碟讀取的位元組數目。 |
| VolumeReadIOPs | <p>來自叢集磁碟區的計費讀取 I/O 操作平均數量，每隔 5 分鐘回報一次。計費的讀取操作以叢集磁碟區層級計算，彙整來自資料庫叢集中的所有執行個體，每隔 5 分鐘回報一次。此值以每隔 5 分鐘取得的讀取操作指標值進行計算。您可以藉由取得計費讀取操作指標並除以 300 秒，計算出每秒的計費讀取操作量。</p> <p>例如，如果 VolumeReadIOPs 傳回 13,686，則計費的讀取作業每秒為 45 ($13,686/300 = 45.62$)。</p> <p>您的查詢提出資料庫頁面請求，該頁面不存在於緩衝區快取，因此必須從儲存空間載入，因而產生計費讀取操作。您可能會在計費讀取操作中發現峰值，那是因為從儲存空間讀取查詢結果，然後載入至緩衝區快取。</p> |

| 指標 | 描述 | |
|-----------------|--|--|
| VolumeWriteIOPs | <p>來自叢集磁碟區的計費寫入 I/O 操作平均數量，每隔 5 分鐘回報一次。計費的寫入操作以叢集磁碟區層級計算，彙整來自資料庫叢集中的所有執行個體，每隔 5 分鐘回報一次。此值以每隔 5 分鐘取得的寫入操作指標值進行計算。您可以藉由取得計費寫入操作指標值並除以 300 秒，計算出每秒的計費寫入操作量。</p> <p>例如，如果 VolumeWriteIOPs 傳回 13,686，則每秒計費的寫入作業為 45 (13,686/300 = 45.62)。</p> <p>請注意 VolumeReadIOPs 和 VolumeWriteIOPs 指標是由 DocumentDB 儲存層計算，其中包含主要執行個體和複本執行個體所執行的 iOS。資料會每 20-30 分鐘彙總一次，然後每 5 分鐘一次報告一次，因此在期間內為指標發出相同的資料點。如果您要尋找在 1 分鐘間隔內與插入作業產生關聯的量度，則可以使用執行個體層級 WriteIOPS 量度。該指標可在 Amazon DocumentDB 主要執行個體的監控索引標籤中找到。</p> | |

| 指標 | 描述 | |
|-----------------|--|--|
| WriteIOPS | 磁碟寫入輸入/輸出操作的每秒平均次數。在叢集層級使用時，WriteIOPs 會針對叢集中的所有執行個體進行評估。每隔 1 分鐘分別回報讀取與寫入 IOPS。 | |
| WriteThroughput | 平均每秒寫入磁碟的位元組數目。 | |

系統

| 指標 | 描述 | |
|--------------------------|---|--|
| BufferCacheHitRatio | 由緩衝區快取提供服務的請求的百分比。 | |
| DiskQueueDepth | 分散式儲存磁碟區的並行寫入要求數目。 | |
| EngineUptime | 執行個體已執行的時間量 (單位秒)。 | |
| IndexBufferCacheHitRatio | 緩衝區快取所服務的索引要求百分比。刪除索引、集合或資料庫之後，您可能會立即看到量度大於 100% 的尖峰。這將在 60 秒後自動更正。此限制將在未來的修補程式更新中修正。 | |

T3 執行處理測量值

| 指標 | 描述 |
|--------------------------|---|
| CPUCreditUsage | 測量期間花費的 CPU 點數。 |
| CPUCreditBalance | 執行處理累積的 CPU 積分數目。當 CPU 效能瞬間突增，以及 CPU 額度用掉的速度比獲得的快時，此額度餘額會用完。 |
| CPUSurplusCreditBalance | 剩餘的 CPU 積分花費在 CPU 時維持 CPU 性能的數量CreditBalance 值為零。 |
| CPUSurplusCreditsCharged | 過剩 CPU 積分超過 24 小時內可獲得的最大 CPU 積分數量，從而產生額外費用。如需詳細資訊，請參閱 監控您的 CPU 積分 。 |

檢視 CloudWatch 資料

您可以查看亞馬遜 CloudWatch 資料:使用 CloudWatch 控制台，亞馬遜文檔數據庫控制台，AWS Command Line Interface(AWS CLI)，或 CloudWatch API。

Using the AWS Management Console

若要檢視 CloudWatch 使用 Amazon 文件資料庫管理主控台完成以下步驟的指標。

1. 登入AWS Management Console，然後在以下位置打開亞馬遜文檔數據庫控制台<https://console.aws.amazon.com/docdb>。
2. 在導覽窗格中，選擇 Clusters (叢集)。

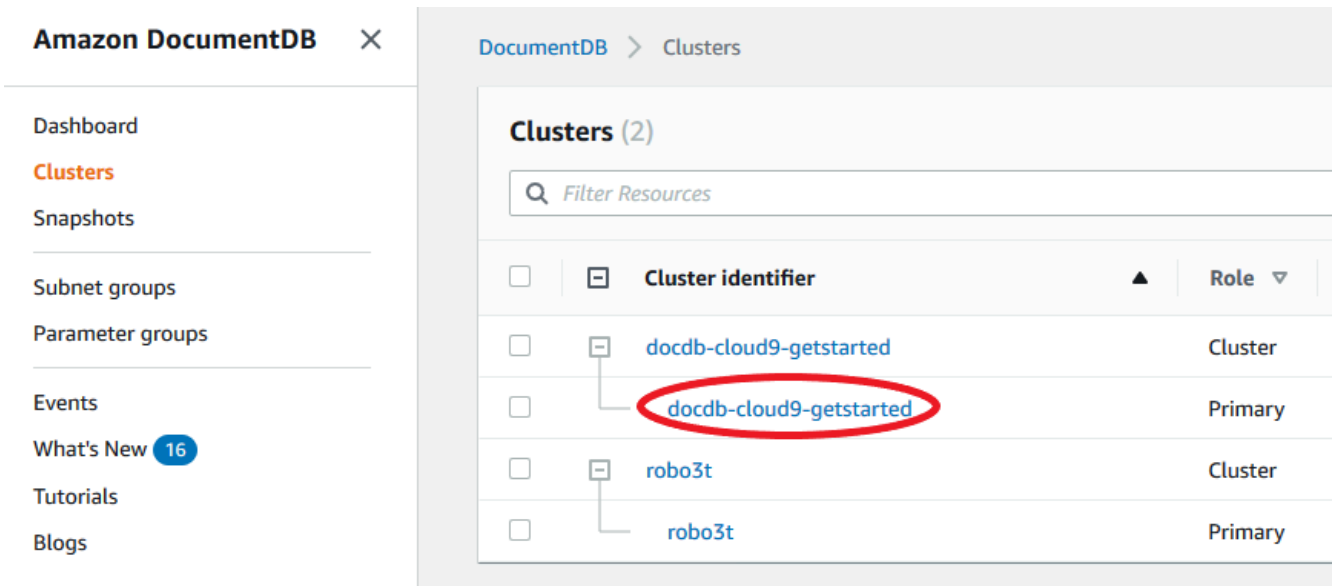
Tip

如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示

(☰)

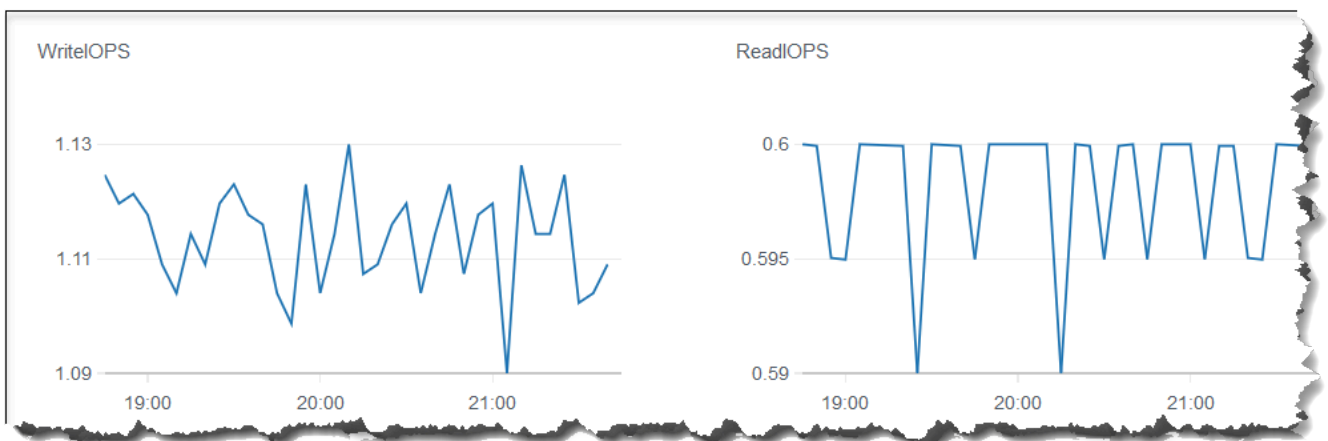
)。

3. 在 [叢集] 導覽方塊中，您會看到該欄叢集識別碼。您的執行個體會列在叢集下，類似下方的螢幕擷取畫面。



4. 從執行個體清單中，選擇您要指標的執行個體名稱。
5. 在產生的執行個體摘要頁面中，選擇監控此索引標籤可檢視 Amazon 文件資料庫執行個體指標的圖形表示。由於必須為每個量度產生圖表，因此可能需要幾分鐘的時間 CloudWatch 要填入的圖形。

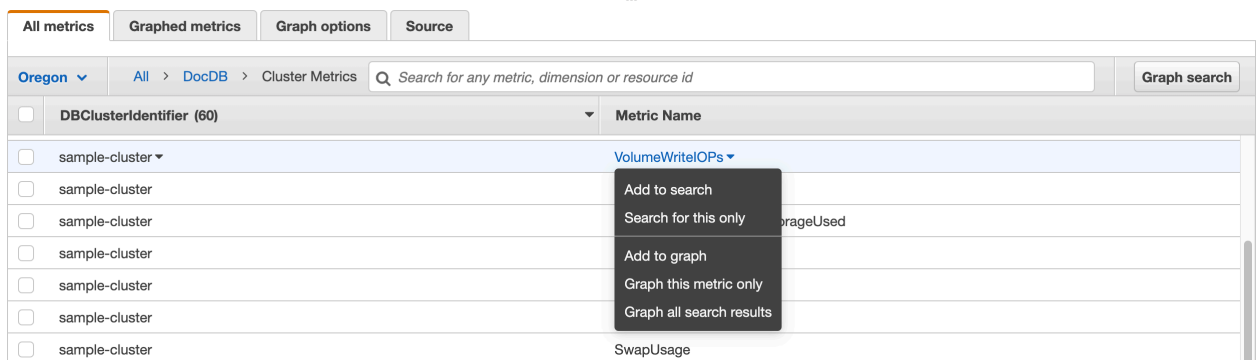
下圖顯示了兩個的圖形表示 CloudWatch 亞馬遜文檔數據庫控制台中的指標，WriteIOPS和ReadIOPS。



Using the CloudWatch Management Console

若要檢視 CloudWatch 量度使用 CloudWatch 管理主控台，請完成下列步驟。

1. 登入AWS Management Console，然後在以下位置打開亞馬遜文檔數據庫控制台<https://console.aws.amazon.com/cloudwatch>。
2. 在導覽窗格中，選擇 Metrics (指標)。然後，從服務名稱清單中選擇文件資料庫。
3. 選擇量度維度 (例如，叢集度量)。
4. 該所有量度標籤會顯示該維度的所有測量結果文件資料庫。
 - a. 若要將資料表排序，請使用直欄標題。
 - b. 若要將指標圖形化，請勾選指標旁的核取方塊。若要選擇所有指標，請勾選表格標題列中的核取方塊。
 - c. 若要依量度篩選，請將游標暫留在量度名稱上，然後選取量度名稱旁邊的下拉式箭頭。然後，選擇添加到搜索，如下圖所示。



Using the AWS CLI

若要檢視 CloudWatch 亞馬遜文檔數據庫的數據，請使用 CloudWatch `get-metric-statistics` 使用以下參數進行操作。

參數

- **--namespace** – 必要。您想要其 CloudWatch 指標的服務命名空間。對於亞馬遜文檔數據庫，這必須是 `AWS/DocDB`。
- **--metric-name** – 必要。您要資料的測量結果名稱。
- **--start-time** – 必要。決定要傳回的第一個資料點的時間戳記。

包含指定的值；結果包括具有指定時間戳記的資料點。時間戳記必須採用 ISO 8601 UTC 格式 (例如 `2016-10-03T23:00:00Z`)。

- **--end-time** – 必要。決定要傳回的最後一個資料點的時間戳記。

包含指定的值；結果包括具有指定時間戳記的資料點。時間戳記必須採用 ISO 8601 UTC 格式 (例如 2016-10-03T23:00:00Z)。

- **--period** – 必要。所傳回資料點的精細程度 (單位秒)。若是使用一般解析度的指標，期間最短可以是一分鐘 (60 秒) 且必須是 60 的倍數。對於依少於一分鐘的間隔收集的高解析度指標，期間可以是 1、5、10、30、60，或 60 的任何倍數。
- **--dimensions**-可選。如果量度包含多個維度，您必須為每個維度包含一個值。CloudWatch 將每個唯一維度組合視為單獨的量度。如果未發佈特定維度組合，您就無法擷取其統計資料。您必須指定建立指標時所使用的相同維度。
- **--statistics**-可選。百分位數以外的指標統計資料。如需百分位數統計資料，請使用 `ExtendedStatistics`。呼叫 `GetMetricStatistics` 時，您必須指定 `Statistics` 或 `ExtendedStatistics`，但不能同時指定兩者。

允許的值：

- `SampleCount`
- `Average`
- `Sum`
- `Minimum`
- `Maximum`
- **--extended-statistics**-可選。percentile 統計資料。指定介於 p0.0 至 p100 之間的值。呼叫 `GetMetricStatistics` 時，您必須指定 `Statistics` 或 `ExtendedStatistics`，但不能同時指定兩者。
- **--unit**-可選。特定指標的單位。指標可能採用多種單位回報。若未提供單位，則會導致傳回所有單位的結果。如果您唯一指定的單位是指標不會報告的單位，則呼叫結果會是 `null`。

可能的值如下：

- `Seconds`
- `Microseconds`
- `Milliseconds`
- `Bytes`
- `Kilobytes`
- `Megabytes`
- `Gigabytes`

- Terabytes
- Bits
- Kilobytes
- Megabits
- Gigabits
- Terabits
- Percent
- Count
- Bytes/Second
- Kilobytes/Second
- Megabytes/Second
- Gigabytes/Second
- Terabytes/Second
- Bits/Second
- Kilobits/Second
- Megabits/Second
- Gigabits/Second
- Terabits/Second
- Count/Second
- None

Example

以下範例會找出 2 小時期間的最大 CPUUtilization，每 60 秒取樣一次。

若為 Linux、macOS 或 Unix：

```
aws cloudwatch get-metric-statistics \  
  --namespace AWS/DocDB \  
  --dimensions \  
    Name=DBInstanceIdentifier,Value=docdb-2019-01-09-23-55-38 \  
  --metric-name CPUUtilization \  
  --start-time 2019-02-11T05:00:00Z \  
  --end-time 2019-02-11T07:00:00Z \  
  --query 'Metrics[0].Maximum'
```



```
--period 60 \  
--statistics Maximum
```

針對 Windows :

```
aws cloudwatch get-metric-statistics ^  
  --namespace AWS/DocDB ^  
  --dimensions ^  
    Name=DBInstanceIdentifier,Value=docdb-2019-01-09-23-55-38 ^  
  --metric-name CPUUtilization ^  
  --start-time 2019-02-11T05:00:00Z ^  
  --end-time 2019-02-11T07:00:00Z ^  
  --period 60 ^  
  --statistics Maximum
```

此操作的輸出如下所示。

```
{  
  "Label": "CPUUtilization",  
  "Datapoints": [  
    {  
      "Unit": "Percent",  
      "Maximum": 4.49152542374361,  
      "Timestamp": "2019-02-11T05:51:00Z"  
    },  
    {  
      "Unit": "Percent",  
      "Maximum": 4.250000000000485,  
      "Timestamp": "2019-02-11T06:44:00Z"  
    },  
    ***** some output omitted for brevity *****  
    {  
      "Unit": "Percent",  
      "Maximum": 4.333333333331878,  
      "Timestamp": "2019-02-11T06:07:00Z"  
    }  
  ]  
}
```

亞馬遜文檔數據庫尺寸

Amazon 文件資料庫的指標是由帳戶或作業的值來限定。您可以使用 CloudWatch 用於擷取下表中任何維度篩選的 Amazon DocumentDB 資料的主控台。

| 維度 | 描述 |
|---------------------------|---|
| DBClusterIdentifier | 篩選您要求特定 Amazon 文件資料庫叢集的資料。 |
| DBClusterIdentifier, Role | 篩選您要求特定 Amazon DocumentDB 叢集的資料，並依執行個體角色彙總指標 (寫入器/讀取器)。例如，您可以為屬於某個叢集的所有讀取者執行個體彙總指標。 |
| DBInstanceIdentifier | 篩選您對特定資料庫執行個體請求的資料。 |

監視反應

對於閒置叢集，運算計數器度量具有非零值 (通常約 50)。這是因為 Amazon DocumentDB 會執行定期運作狀態檢查、內部操作和指標收集任務。

監控資料庫連線

當您使用資料庫引擎命令 (例如) 來檢視連線數目時 `db.runCommand({ serverStatus: 1 })`，您可能會看到比您在中看到的多 10 個連線 `DatabaseConnections` 通過 CloudWatch。發生這種情況的原因是 Amazon DocumentDB 執行定期運作狀態檢查和指標收集任務，而這些任務不會被納入 `DatabaseConnections`。 `DatabaseConnections` 僅代表客戶啟動的連線。

使用記錄 Amazon DocumentDB API 呼叫 AWS CloudTrail

Amazon DocumentDB (相容於 MongoDB 相容性) 整合了 AWS CloudTrail，後者是一項服務，可提供由使用者、角色或 AWS 服務在 Amazon DocumentDB 中所採取之動作的記錄。CloudTrail 將 Amazon DocumentDB 的所有 AWS CLI API 呼叫擷取為事件，包括來自 Amazon DocumentDB 主控台的呼叫以及來自對 Amazon DocumentDB SDK SDK SDK 發出的程式碼呼叫。如果您建立線索，就可以持續傳送 CloudTrail 事件至 Amazon S3 儲存貯體，包括 Amazon DocumentDB 的事件。即使未設定追蹤，您依

然可以在 CloudTrail 主控台的事件歷史記錄中檢視最新事件。您可利用所 CloudTrail 收集的資訊來判斷向 Amazon DocumentDB 發出的請求，以及發出請求的 IP 地址、人員、時間和其他詳細資訊。

Important

對於某些管理功能，Amazon DocumentDB 使用與 Amazon Relational Database Service (Amazon RDS) 共享的操作技術。Amazon DocumentDB 主控 AWS CLI 台和 API 呼叫會記錄為對亞馬遜 RDS API 的呼叫。

若要進一步了解 AWS CloudTrail，請參閱 [《AWS CloudTrail 使用者指南》](#)。

Amazon DocumentDB 信息 CloudTrail

CloudTrail 當您建立帳戶 AWS 帳戶時，系統即會在中啟用。當 Amazon DocumentDB 中發生活動時 (與 MongoDB 相容性) 中發生活動時，該活動便會與其他 AWS 服務 CloudTrail 事件一同記錄在事件歷史記錄中。您可以檢視、搜尋和下載 AWS 帳戶 的最新事件。如需詳細資訊，請參閱 [使用 CloudTrail 事件歷史記錄檢視事件](#)。

若要持續記錄您的事件 AWS 帳戶，包括 Amazon DocumentDB 的事件 (與 MongoDB 相容性) 的事件 (與 MongoDB 相容性)，請建立追蹤。線索能 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱 AWS CloudTrail 使用者指南中的以下主題：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務與整合](#)
- [設定的 Amazon SNS SNS SNS SNS SNS SNS SNS SNS CloudTrail](#)
- [接收多個區域的 CloudTrail 日誌檔案](#)
- [接收多個帳戶的 CloudTrail 日誌檔案](#)

每一筆事件或記錄項目都會包括產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail 使用者身分元素](#)。

分析亞馬遜文檔數據庫操作

您可以使用 Amazon DocumentDB 中的效能分析工具 (與 MongoDB 相容性) 來記錄叢集上執行的作業的執行時間和詳細資料。Profiler 適用於監控叢集上最慢的操作，以協助您改善個別查詢效能和整體叢集效能。

根據預設會停用 Profiler 功能。啟用時，效能分析工具會將花費超過客戶定義的臨界值 (例如 100 ms) 的操作記錄到 Amazon CloudWatch 記錄檔。記錄的詳細資訊包括已分析的命令、時間、計劃摘要和用戶端中繼資料。操作記錄到之後 CloudWatch 日誌，您可以使用 CloudWatch 日誌深入解析以分析、監控和存檔您的 Amazon DocumentDB 分析資料。[常用查詢](#) 小節提供常用的查詢。

啟用時，Profiler 會使用叢集的額外資源。我們建議您從高閾值 (例如 500 毫秒) 開始，再逐漸降低該值，以識別慢速操作。對於高輸送量應用程式，從閾值 50 毫秒開始可能會在叢集上造成效能問題。效能分析工具會在叢集層級啟用，並可用於叢集中的所有執行個體和資料庫。亞馬遜文檔數據庫日誌操作到亞馬遜 CloudWatch 在最大努力的基礎上記錄。

雖然 Amazon DocumentDB 不會收取額外費用來啟用效能分析工具，但是我們會向您收取使用量的標準費率 CloudWatch 記錄檔。如需相關資訊 CloudWatch 記錄檔定價，請參閱 [亞馬遜 CloudWatch 價錢](#)。

主題

- [受支援的操作](#)
- [限制](#)
- [啟用亞馬遜文件資料庫效能分析工具](#)
- [停用亞馬遜文件資料庫效能分析工具](#)
- [停用 Profiler 日誌匯出](#)
- [存取您的亞馬遜文件資料庫效能分析工具日誌](#)
- [常用查詢](#)

受支援的操作

亞馬遜文檔數據庫分析工具支持以下操作：

- aggregate
- count

- delete
- distinct
- find (OP_QUERY 和命令)
- findAndModify
- insert
- update

限制

只有在查詢的整個結果集能夠容納一個批次，且結果集小於 16MB (BSON 大小上限) 時，慢速查詢效能分析工具才能發出效能分析工具記錄。大於 16MB 的結果集會自動分割成多個批次。

大多數驅動程式或 shell 可能會設定較小的預設批次大小。您可以將批次大小指定為查詢的一部分。為了捕獲緩慢的查詢日誌，我們建議使用超過預期結果集大小的批次大小。如果您不確定結果集大小，或者結果有所不同，也可以將批次大小設定為較大的數字 (例如，100k)。

但是，使用較大的批次大小意味著在將回應傳送給用戶端之前，必須從資料庫擷取更多結果。對於某些查詢，這可能會在獲得結果之前產生更長的延遲。如果您不打算使用整個結果集，則可能會花費更多 I/O 來處理查詢並丟棄結果。

啟用亞馬遜文件資料庫效能分析工具

在叢集上啟用 Profiler 包含三個步驟。確定所有步驟都已完成，否則效能分析記錄檔將不會傳送至 CloudWatch 記錄檔。Profiler 是在叢集層級設定，並在叢集的所有資料庫和執行個體上執行。

在叢集上啟用 Profiler

1. 由於您無法修改預設叢集參數群組，請確保您有可用的自訂叢集參數群組。如需詳細資訊，請參閱[建立 Amazon DocumentDB 叢集參數群組](#)。
2. 使用可用的自訂叢集參數群組，修改下列參數：profiler、profiler_threshold_ms 和 profiler_sampling_rate。如需詳細資訊，請參閱[修改 Amazon DocumentDB 叢集參數群組](#)。
3. 建立或修改叢集以使用自訂叢集參數群組，並啟用匯出 profiler 記錄到 CloudWatch 記錄檔。

以下章節說明如何使用 AWS Management Console 和 AWS Command Line Interface (AWS CLI) 實作這些步驟。

Using the AWS Management Console

1. 在開始之前，請先建立 Amazon DocumentDB 叢集和自訂叢集參數群組 (如果您還沒有叢集參數群組)。如需詳細資訊，請參閱 [建立 Amazon DocumentDB 叢集參數群組](#) 及 [創建一個 Amazon DocumentDB 叢集](#)。
2. 使用可用的自訂叢集參數群組，修改下列參數。如需詳細資訊，請參閱 [修改 Amazon DocumentDB 叢集參數群組](#)。
 - `profiler`— 啟用或停用查詢效能分析。允許值為 `enabled` 和 `disabled`。預設值為 `disabled`。若要啟用分析，請將值設定為 `enabled`。
 - `profiler_threshold_ms`— 何時 `profiler` 設定為 `enabled`，所有使用時間超過 `profiler-threshold-ms` 會記錄到 CloudWatch。允許值為 `[50-INT_MAX]`。預設值為 `100`。
 - `profiler_sampling_rate`— 應進行效能分析或記錄的緩慢作業的部分。允許值為 `[0.0-1.0]`。預設值為 `1.0`。
3. 修改叢集以使用自訂叢集參數群組，並將效能分析工具日誌匯出設定為發佈到 Amazon CloudWatch。
 - a. 在導覽窗格中，選擇 Clusters (叢集) 來將自訂參數群組新增至叢集。
 - b. 選擇參數群組要與其建立關聯之叢集名稱左側的按鈕。選取 Actions (動作)，然後選取 Modify (修改) 以修改您的叢集。
 - c. 在 Cluster (叢集) 選項下，從上述步驟中選擇自訂參數群組，以將其新增至您的叢集。
 - d. 下記錄匯出，選取效能分析工具記錄檔發布到亞馬遜 CloudWatch。
 - e. 選擇 Continue (繼續) 以檢視修改摘要。
 - f. 確認變更後，即可立即套用，或是在下一個維護時段在 Scheduling of modifications (修改的排程) 下套用。
 - g. 選擇 Modify cluster (修改叢集) 來以新的參數群組更新叢集。

Using the AWS CLI

下列程序會在叢集 `sample-cluster` 的所有受支援操作上啟用 Profiler。

1. 在開始之前，請執行下列命令，並檢閱名稱中沒有 `default` 且具有 `docdb3.6` 作為參數群組系列之叢集參數群組的輸出，以確定您擁有可用的自訂叢集參數群組。如果您沒有非預設的叢集參數群組，請參閱 [建立 Amazon DocumentDB 叢集參數群組](#)。

```
aws docdb describe-db-cluster-parameter-groups \
  --query 'DBClusterParameterGroups[*].
  [DBClusterParameterGroupName,DBParameterGroupFamily]'
```

在下列輸出中，僅 `sample-parameter-group` 符合這兩個條件。

```
[
  [
    "default.docdb3.6",
    "docdb3.6"
  ],
  [
    "sample-parameter-group",
    "docdb3.6"
  ]
]
```

2. 使用您的自訂叢集參數群組，修改下列參數：

- `profiler`— 啟用或停用查詢效能分析。允許值為 `enabled` 和 `disabled`。預設值為 `disabled`。若要啟用分析，請將值設定為 `enabled`。
- `profiler_threshold_ms`— 何時 `profiler` 設定為 `enabled`，所有命令花費的時間超過 `profiler -threshold-ms` 會記錄到 CloudWatch。允許值為 `[0-INT_MAX]`。將此值設定為 `0` 會分析所有支援的操作。預設值為 `100`。
- `profiler_sampling_rate`— 應進行效能分析或記錄的緩慢作業的部分。允許值為 `[0.0-1.0]`。預設值為 `1.0`。

```
aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name sample-parameter-group \
  --parameters
  ParameterName=profiler,ParameterValue=enabled,ApplyMethod=immediate \
  ParameterName=profiler_threshold_ms,ParameterValue=100,ApplyMethod=immediate \
  ParameterName=profiler_sampling_rate,ParameterValue=0.5,ApplyMethod=immediate
```

3. 修改您的亞馬遜文檔數據庫叢集，使其使用 `sample-parameter-group` 自定義集群參數組從上一步，並設置參數 `--enable-cloudwatch-logs-exports` 至 `profiler`。

下面的代碼修改集群sample-cluster若要使用sample-parameter-group從上一步，並添加profiler到已啟用 CloudWatch 記錄匯出。

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-parameter-group-name sample-parameter-group \  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["profiler"]}'
```

此操作的輸出將會如下所示。

```
{  
  "DBCluster": {  
    "AvailabilityZones": [  
      "us-east-1c",  
      "us-east-1b",  
      "us-east-1a"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DBClusterIdentifier": "sample-cluster",  
    "DBClusterParameterGroup": "sample-parameter-group",  
    "DBSubnetGroup": "default",  
    "Status": "available",  
    "EarliestRestorableTime": "2020-04-07T02:05:12.479Z",  
    "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",  
    "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "docdb",  
    "EngineVersion": "3.6.0",  
    "LatestRestorableTime": "2020-04-08T22:08:59.317Z",  
    "Port": 27017,  
    "MasterUsername": "test",  
    "PreferredBackupWindow": "02:00-02:30",  
    "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",  
    "DBClusterMembers": [  
      {  
        "DBInstanceIdentifier": "sample-instance-1",  
        "IsClusterWriter": true,  
        "DBClusterParameterGroupStatus": "in-sync",  
        "PromotionTier": 1  
      },  
      {
```



```
        "DBInstanceIdentifier": "sample-instance-2",
        "IsClusterWriter": true,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
    }
],
"VpcSecurityGroups": [
    {
        "VpcSecurityGroupId": "sg-abcd0123",
        "Status": "active"
    }
],
"HostedZoneId": "ABCDEFGHIJKLM",
"StorageEncrypted": true,
"KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
"DbClusterResourceId": "cluster-ABCDEFGHIJKLMNQRSTUWXYZ",
"DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-
cluster",
"AssociatedRoles": [],
"ClusterCreateTime": "2020-01-10T22:13:38.261Z",
"EnabledCloudwatchLogsExports": [
    "profiler"
],
"DeletionProtection": true
}
}
```

停用亞馬遜文件資料庫效能分析工具

若要停用效能分析工具，請同時停用 `profiler` 參數和導出 `profiler` 記錄到 CloudWatch 記錄檔。

停用 Profiler

您可以使用 AWS Management Console 或 AWS CLI 來停用 `profiler` 參數，如下所示。

Using the AWS Management Console

下列程序會使用 AWS Management Console 禁用亞馬遜文檔數據庫 `profiler`。

1. 登入 AWS Management Console，然後在以下位置打開亞馬遜文檔數據庫控制台 <https://console.aws.amazon.com/docdb>。

2. 在導覽窗格中，選擇 Parameter groups (參數群組)。然後選擇您要停用 Profiler 的叢集參數群組的名稱。
3. 在產生的 Cluster parameters (叢集參數) 頁面中，選取 profiler 參數左邊的按鈕，然後選擇 Edit (編輯)。
4. 在 Modify profiler (修改 Profiler) 對話方塊中，選擇清單中的 disabled。
5. 選擇 Modify cluster parameter (修改叢集參數)。

Using the AWS CLI

若要使用 AWS CLI 在叢集上停用 profiler，請修改叢集，如下所示。

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group \  
  --parameters  
  ParameterName=profiler,ParameterValue=disabled,ApplyMethod=immediate
```

停用 Profiler 日誌匯出

您可以禁用導出 profiler 記錄到 CloudWatch 使用記錄 AWS Management Console 或者 AWS CLI，如下所示。

Using the AWS Management Console

下列程序會使用 AWS Management Console 禁用亞馬遜文檔數據庫將日誌導出到 CloudWatch。

1. 打開亞馬遜文檔數據庫控制台 <https://console.aws.amazon.com/docdb>。
2. 在導覽窗格中，選擇 Clusters (叢集)。對於您要停止匯出日誌的叢集，選擇其名稱左側的按鈕。
3. 在 Actions (動作) 功能表上，選擇 Modify (修改)。
4. 向下捲動至 Log exports (日誌匯出) 區段，並取消選取 Profiler logs (分析工具日記)。
5. 選擇 Continue (繼續)。
6. 檢閱您的變更，然後選擇希望叢集套用此變更的時機：
 - Apply during the next scheduled maintenance window (在下次排定的維護時段套用)
 - Apply immediately (立即套用)
7. 選擇 Modify cluster (修改叢集)。

Using the AWS CLI

下面的代碼修改集群 `sample-cluster` 並停用 CloudWatch 效能分析工具記錄檔。

Example

若為 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

針對 Windows：

```
aws docdb modify-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

此操作的輸出將會如下所示。

```
{  
  "DBCluster": {  
    "AvailabilityZones": [  
      "us-east-1c",  
      "us-east-1b",  
      "us-east-1a"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DBClusterIdentifier": "sample-cluster",  
    "DBClusterParameterGroup": "sample-parameter-group",  
    "DBSubnetGroup": "default",  
    "Status": "available",  
    "EarliestRestorableTime": "2020-04-08T02:05:17.266Z",  
    "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",  
    "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "docdb",  
    "EngineVersion": "3.6.0",  
    "LatestRestorableTime": "2020-04-09T05:14:44.356Z",  
    "Port": 27017,  
    "MasterUsername": "test",  
    "PreferredBackupWindow": "02:00-02:30",  
    "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
```

```
"DBClusterMembers": [
  {
    "DBInstanceIdentifier": "sample-instance-1",
    "IsClusterWriter": true,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "sample-instance-2",
    "IsClusterWriter": true,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  }
],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-abcd0123",
    "Status": "active"
  }
],
"HostedZoneId": "ABCDEFGHIJKLM",
"StorageEncrypted": true,
"KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
"DbClusterResourceId": "cluster-ABCDEFGHIJKLMNQRSTUWXYZ",
"DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster",
"AssociatedRoles": [],
"ClusterCreateTime": "2020-01-10T22:13:38.261Z",
"DeletionProtection": true
}
}
```

存取您的亞馬遜文件資料庫效能分析工具日誌

請按照以下步驟訪問您在 Amazon 上的個人資料日誌 CloudWatch。

1. 打開 CloudWatch 控制台 <https://console.aws.amazon.com/cloudwatch/>。
2. 請確定您所在的區域與 Amazon 文件資料庫叢集位於相同的區域。
3. 在導覽窗格中，選擇 Logs (日誌)。
4. 若要尋找叢集的分析工具日誌，請在清單中選擇 `/aws/docdb/yourClusterName/profiler`。

您可在每個執行個體名稱下分別取得每個執行個體的描述檔日誌。

常用查詢

以下是一些常用的查詢，可用來分析您已側錄的命令。有關更多信息 CloudWatch 日誌洞察，請參閱[分析記錄資料 CloudWatch 日誌洞察](#)和[範例查詢](#)。

取得指定集合上最慢的 10 個操作

```
filter ns="test.foo" | sort millis desc | limit 10
```

取得集合上耗費超過 60 毫秒的所有更新操作

```
filter millis > 60 and op = "update"
```

取得上個月最慢的 10 個操作

```
sort millis desc | limit 10
```

使用 COLLSCAN 計劃摘要取得所有查詢

```
filter planSummary="COLLSCAN"
```

使用績效詳情進行監控

Performance Insights 新增至現有的 Amazon DocumentDB 監控功能，以說明叢集效能並協助您分析影響叢集的任何問題。透過「Performance Insights」儀表板，您可以視覺化呈現資料庫負載，並依等待、查詢陳述式、主機或應用程式篩選負載。

Note

Performance Insights 僅適用於 Amazon DocumentDB 3.6、4.0 和 5.0 執行個體型叢集。

它是如何有用的？

- 視覺化資料庫效能 — 以視覺化方式呈現負載，以判斷負載在資料庫上的時間和位置
- 判斷造成資料庫負載的原因 — 判斷哪些查詢、主機和應用程式會導致執行個體的負載
- 判斷資料庫何時有負載 — 放大 Performance Insights 儀表板以專注於特定事件，或縮小以查看更大時間範圍內的趨勢
- 資料庫負載警示 — 自動存取新的資料庫載入指標，您 CloudWatch 可以從中監視資料庫載入指標以及其他 DocumentDB 指標，並在其上設定警示

Amazon DocumentDB Performance Insights 有哪些限制？

- 尚未提供 AWS GovCloud (美國西部) 區域的 Performance Insights
- DocumentDB 的 Performance Insights 可保留長達 7 天的效能資料
- 長度超過 1024kb 的查詢不會彙總在 Performance Insights

主題

- [Performance Insights 概念](#)
- [啟用和停用績效詳情](#)
- [設定績效詳情的存取政策](#)
- [使用績效詳情儀表板來分析指標](#)
- [使用績效詳情 API 來擷取指標](#)
- [亞馬遜 CloudWatch 指標的 Performance Insights](#)
- [計數器指標的 Performance Insights](#)

Performance Insights 概念

主題

- [平均作用中工作階段](#)
- [維度](#)
- [最大 vCPU](#)

平均作用中工作階段

資料庫負載 (DB 負載) 測量資料庫中的活動程度。績效詳情中的關鍵指標為 DB Load，每秒收集一次。DBLoad測量結果的單位是 DocumentDB 執行處理的「平均作用中階段作業 (AAS)」。

使用中工作階段是已將工作提交至 DocumentDB 執行個體並等待回應的連線。例如，如果您將查詢提交至 DocumentDB 執行個體，則執行個體正在處理查詢時，資料庫工作階段會處於作用中狀態。

若要取得平均作用中工作階段 (AAS)，績效詳情會取樣同時執行查詢的工作階段數目。AAS 是工作階段總數除以樣本總數。下表顯示執行中查詢的五個連續範例。

| 樣本 | 執行查詢的工作階段數目 | AAS | 算式 |
|----|-------------|-----|------------------|
| 1 | 2 | 2 | 2 個工作階段 / 1 個樣本 |
| 2 | 0 | 1 | 2 個工作階段 / 2 個樣本 |
| 3 | 4 | 2 | 6 個工作階段 / 3 個樣本 |
| 4 | 0 | 1.5 | 6 個工作階段 / 4 個樣本 |
| 5 | 4 | 2 | 10 個工作階段 / 5 個樣本 |

在上述範例中，從 1-5 之間的時間間隔的「資料庫負載」為 2 AAS。資料庫負載的增加意味著，就平均而言，更多的工作階段正執行於資料庫上。

維度

DB Load 指標與其他時間序列指標不同，因為您可以將它分為名為維度的子元件。您可以將維度視為 DB Load 指標的不同特性類別。當您診斷效能問題時，最有用的維度是等待狀態和最常用的查詢。

等待狀態

等待狀態會導致查詢陳述式等待特定事件發生，然後才能繼續執行。例如，查詢陳述式可能會等到鎖定的資源解除鎖定為止。通過 DB Load 與等待狀態結合，您可以獲得會話狀態的完整圖片。以下是各種 DocumentDB 等待狀態：

| DocumentDB 等待狀態 | 等待狀態說明 |
|-----------------|--|
| 門鎖 | 當工作階段正在等待緩衝區集區分頁時，就會發生 Locker 等待狀態。當系統經常處理大型查詢、集合掃描，或緩衝區集區太小而無法處理工作集時，頻繁進出緩衝區集區的分頁可能會更頻繁地發生。 |
| CPU | 當工作階段正在等待 CPU 時，就會發生 CPU 等待狀態。 |
| CollectionLock | CollectionLock 等待狀態會在工作階段等候取得集合上的鎖定時發生。當集合上有 DDL 作業時，就會發生這些事件。 |
| DocumentLock | DocumentLock 等待狀態會在工作階段等待取得文件上的鎖定時發生。同一份文件的大量並行寫入將有助於在該文件上產生更多的 DocumentLock 等待狀態。 |
| SystemLock | SystemLock 等待狀態會在工作階段在系統上等待時發生。當系統上有頻繁的長時間運行的查詢，長時間運行的事務或高並發性時，可能會發生這種情況。 |
| IO | IO 等待狀態會在等待 IO 上完成的工作階段時發生。 |
| BufferLock | 當工作階段正在 BufferLock 等待取得緩衝區中公用頁面上的鎖定時，就會發生等待狀態。BufferLock 如果其他進程在請求的頁面上保留打開的游標，則等待狀態可以延長。 |
| LowMemThrottle | 當工作階段因為 Amazon DocumentDB 執行個體上的記憶體壓力過高而 LowMemThrottle 等待時，就會發生等待狀態。如果此狀態持續很長一段時間，請考慮擴展執行個體以提供額外的記憶體。如需詳細資訊，請參閱 資源管理員 。 |

| DocumentDB 等待狀態 | 等待狀態說明 |
|--------------------|---|
| BackgroundActivity | 當工作階段正在 BackgroundActivity 等待內部系統處理序時，就會發生等待狀態。 |
| 其他 | 其他等待狀態是內部等待狀態。如果此狀態持續很長一段時間，請考慮終止此查詢。如需詳細資訊，請參閱 如何尋找和終止長時間執行或封鎖的查詢？ |

熱門查詢

等待狀態顯示瓶頸，而頂級查詢顯示哪些查詢對數據庫負載的貢獻最大。例如，許多查詢目前可能正在資料庫上執行，但單一查詢可能會耗用 99% 的資料庫負載。在此情況下，高負載可能表示查詢發生問題。

最大 vCPU

在儀表中，資料庫負載圖表會收集、彙整並顯示工作階段資訊。若要查看作用中的工作階段是否超過最大 CPU，請查看它與最大 vCPU 數線的關係。vCPU 上限值取決於您的 DocumentDB 執行個體的 vCPU (虛擬 CPU) 核心數目。

若資料庫負載通常高於最大 vCPU 數線，而主要等待狀態為 CPU，則 CPU 會超過負載。在這種情況下，您可能想要限制與執行個體的連線、調整 CPU 負載高的任何查詢，或考慮使用較大的執行個體類別。處於任何等待狀態的密集且穩定的執行個體表示可能有您應解決的瓶頸或資源爭用問題。即使資料庫負載未超過最大 vCPU 數線，仍可能會有上述的問題。

啟用和停用績效詳情

若要使用績效詳情，請在資料庫執行個體上啟用。如有必要，您可以稍後禁用它。啟用和停用績效詳情不會造成停機、重新開機或容錯移轉。

績效詳情代理程式會耗用資料庫主機上有限的 CPU 和記憶體。當資料庫負載過高時，代理程式會減少收集資料的頻率來限制效能影響。

建立叢集時啟用 Performance Insights

在主控台中，您可以在建立或修改新的資料庫執行個體時啟用或停用績效詳情。

使用 AWS Management Console

在主控台中，您可以在建立 DocumentDB 叢集時啟用 Performance Insights。當您建立新的 DocumentDB 叢集時，請選擇 [Performance Insights] 區段中的 [啟用 Performance Insights]，以啟用 Performance Insights。

控制台說明

1. 若要建立叢集，請依照[建立 Amazon DocumentDB 叢集](#)中的指示進行。
2. 選取 [效能洞見] 區段中的 [啟用 Performance Insights]。

Performance Insights [Info](#)

Enable Performance Insights

AWS KMS Key [Info](#)

(default) aws/rds

Account

KMS key ID

 You can't change the KMS key after enabling Performance Insights.

Note

Performance Insights 資料保留期為七天。

AWS KMS金鑰 — 指定您的 AWS KMS 金鑰。績效詳情使用您的 AWS KMS 金鑰來對所有可能具敏感性的資料進行加密。將會對傳輸中與靜態資料進行加密。如需詳細資訊，請參閱設定 Performance Insights 的AWSAWS KMS原則。

修改執行個體時啟用和停用

您可以使用主控台或修改資料庫執行個體以啟用或停用 Performance Insights AWS CLI。

Using the AWS Management Console

控制台說明

1. [登入AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 選擇 Clusters (叢集)。
3. 選擇資料庫執行個體，然後選擇 修改 (Modify)。
4. 在 [Performance Insights] 區段中，選擇 [啟用 Performance Insights] 或 [停用 Performance Insights]。

Note

如果您選擇啟用 Performance Insights，您可以指定AWS KMS金鑰。績效詳情使用您的 AWS KMS 金鑰來對所有可能具敏感性的資料進行加密。將會對傳輸中與靜態資料進行加密。如需詳細資訊，請參閱[加密靜態 Amazon 文件資料庫資料](#)。

5. 選擇 Continue (繼續)。
6. 在 Scheduling of Modifications (修改排程) 中，選擇 Apply immediately (立即套用)。如果您在下一個排定的維護時段期間選擇「套用」，則執行個體會忽略此設定，並立即啟用 Performance Insights。
7. 選擇 Modify instance (修改執行個體)。

Using the AWS CLI

當您使用 `create-db-instance` 或 `modify-db-instance` AWS CLI 命令時，您可以指定或透過指定 `--enable-performance-insights` 來停用 Performance Insights 來啟用效能洞見 `--no-enable-performance-insights`。

下列程序說明如何使用啟用或停用資料庫執行個體的 Performance Insights AWS CLI。

AWS CLI 指示

呼叫 `modify-db-instance` AWS CLI 命令並提供下列值：

- `--db-instance-identifier`— 資料庫執行個體的名稱
- `--enable-performance-insights` 以啟用或 `--no-enable-performance-insights` 以停用

Example

下列範例會啟用下列項目的 Performance Insights sample-db-instance :

For Linux, macOS, or Unix:

```
aws docdb modify-db-instance \  
  --db-instance-identifier sample-db-instance \  
  --enable-performance-insights
```

For Windows:

```
aws docdb modify-db-instance ^\  
  --db-instance-identifier sample-db-instance ^\  
  --enable-performance-insights
```

設定績效詳情的存取政策

若要存取績效詳情，您必須具有來自 AWS Identity and Access Management (IAM) 的適用許可。您可以使用下列選項授予存取權：

- 將 AmazonRDSPerformanceInsightsReadOnly 受管政策連接到許可集或角色。
- 建立自訂 IAM 政策，並將其連接至許可集或角色。

此外，如果您在開啟績效詳情時指定客戶受管金鑰，請確保您帳戶中的使用者具備 KMS 金鑰的 kms:Decrypt 和 kms:GenerateDataKey 許可。

Note

[對於 encryption-at-rest AWS KMS 金鑰和安全群組管理，Amazon DocumentDB 會運用與 Amazon RDS 共用的操作技術。](#)

將亞馬遜 RDS PerformanceInsightsReadOnly 政策附加到 IAM 主體

AmazonRDSPerformanceInsightsReadOnly 這是一項 AWS 受管政策，可授予對 Amazon DocumentDB Performance Insights API 所有唯讀操作的存取權。目前，此 API 中的所有操作都是唯讀操作。若您將 AmazonRDSPerformanceInsightsReadOnly 連接至許可集或角色，收件人可使用績效詳情搭配其他主控台功能。

建立績效詳情的自訂 IAM 政策

對於沒有 AmazonRDSPerformanceInsightsReadOnly 政策提供的使用者，您可以建立或修改由使用者管理的 IAM 政策來授予績效詳情的存取權。當您將原則附加至權限集或角色時，收件者可以使用 Performance Insights。

建立自訂政策

1. 前往網址 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
2. 在導覽窗格中，選擇 政策。
3. 選擇 Create policy (建立政策)。
4. 在 Create Policy (建立政策) 頁面上，選擇 JSON 標籤。
5. 複製並貼上以下文字，將 *us-east-1* 換成您的 AWS 區域名稱，將 *111122223333* 換成您的客戶帳戶號碼。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "rds:DescribeDBInstances",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "rds:DescribeDBClusters",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "pi:DescribeDimensionKeys",
      "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
    },
    {
      "Effect": "Allow",
      "Action": "pi:GetDimensionKeyDetails",
      "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
    },
    {
      "Effect": "Allow",
```

```

    "Action": "pi:GetResourceMetadata",
    "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
  },
  {
    "Effect": "Allow",
    "Action": "pi:GetResourceMetrics",
    "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
  },
  {
    "Effect": "Allow",
    "Action": "pi:ListAvailableResourceDimensions",
    "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
  },
  {
    "Effect": "Allow",
    "Action": "pi:ListAvailableResourceMetrics",
    "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
  }
]
}

```

6. 選擇 Review policy (檢閱政策)。
7. 為政策提供名稱並選擇性輸入描述，然後選擇 Create policy (建立政策)。

現在您可以將政策連接到許可集或角色。以下程序假設您已有基於此用途使用的使用者。

將政策連接至使用者

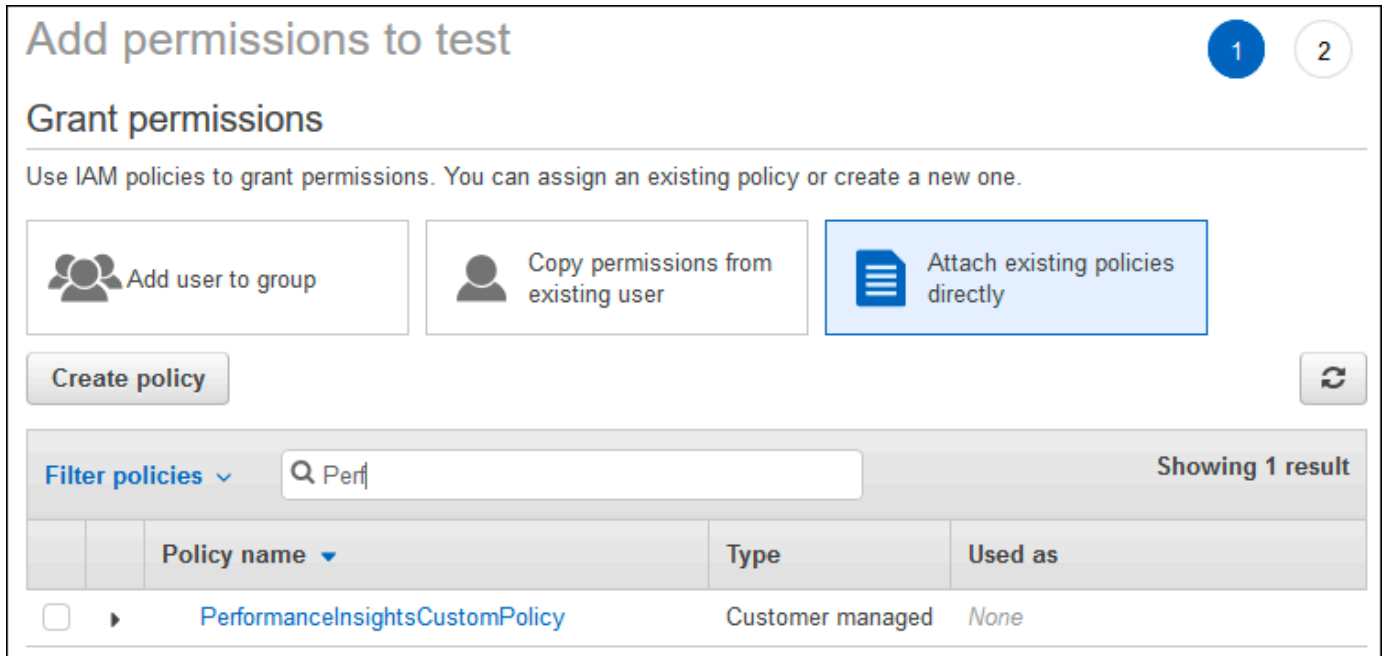
1. 在以下網址開啟 IAM 主控台：<https://console.aws.amazon.com/iam/>。
2. 在導覽窗格中，選擇 Users (使用者)。
3. 從清單中選擇現有的使用者。

Important

若要使用 Performance Insights，請確定除了自訂政策之外，您還可以存取 Amazon DocumentDB。例如，資 AmazonDoc 料庫 ReadOnlyAccess 預先定義的政策提供 Amazon 文件的唯讀存取權。如需詳細資訊，請參閱使用政策 [管理存取](#)。

4. 在 Summary (摘要) 頁面上，選擇 Add permissions (新增許可)。

5. 選擇 **Attach existing policies directly** (直接連接現有政策)。對於 **Search** (搜尋)，請輸入政策名稱的前幾個字母，如下所示。



6. 選擇您的政策，然後選擇 **Next: Review** (下一步：檢閱)。
7. 選擇 **Add permissions** (新增許可)。

設定績效詳情的 AWS KMS 政策

績效詳情使用 AWS KMS key 加密敏感資料。當您透過 API 或主控台啟用績效詳情時，可使用下列選項：

- 選擇預設 AWS 受管金鑰。

Amazon DocumentDB 將用 AWS 受管金鑰於您的新資料庫執行個體。Amazon DocumentDB 為您的帳戶創建 AWS 受管金鑰一個 AWS。您的 AWS 帳戶對於每 AWS 個區域 AWS 受管金鑰的 Amazon DocumentDB 有不同。

- 選擇客戶受管金鑰。

如果您指定客戶受管金鑰，則您帳戶中呼叫績效詳情 API 的使用者需要 KMS 金鑰的 `kms:Decrypt` 和 `kms:GenerateDataKey` 許可。您可以透過 IAM 政策設定這些許可。不過，我們建議您透過 KMS 金鑰政策來管理這些許可。如需詳細資訊，請參閱 [在 AWS KMS 中使用金鑰政策](#)。

Example

下列範例金鑰政策說明如何將陳述式新增至 KMS 金鑰政策。這些陳述式允許存取績效詳情。根據您的使用方式 AWS KMS，您可能需要變更某些限制。在將陳述式新增至您的政策之前，請先移除所有註解。

```
{
  "Version" : "2012-10-17",
  "Id" : "your-policy",
  "Statement" : [ {
    //This represents a statement that currently exists in your policy.
  }
  ....,
  //Starting here, add new statement to your policy for Performance Insights.
  //We recommend that you add one new statement for every RDS/DocumentDB instance
  {
    "Sid" : "Allow viewing RDS Performance Insights",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        //One or more principals allowed to access Performance Insights
        "arn:aws:iam::444455556666:role/Role1"
      ]
    },
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition" :{
      "StringEquals" : {
        //Restrict access to only RDS APIs (including Performance Insights).
        //Replace *region* with your AWS Region.
        //For example, specify us-west-2.
        "kms:ViaService" : "rds.*region*.amazonaws.com"
      },
      "ForAnyValue:StringEquals": {
        //Restrict access to only data encrypted by Performance Insights.
        "kms:EncryptionContext:aws:pi:service": "rds",
        "kms:EncryptionContext:service": "pi",

        //Restrict access to a specific DocDB instance.
        //The value is a DbResourceID.
      }
    }
  }
]
```



```
        "kms:EncryptionContext:aws:rds:db-id": "db-AAAAABBBBBCCCCDDDDDEEEEEE"  
    }  
}  
}
```

使用績效詳情儀表板來分析指標

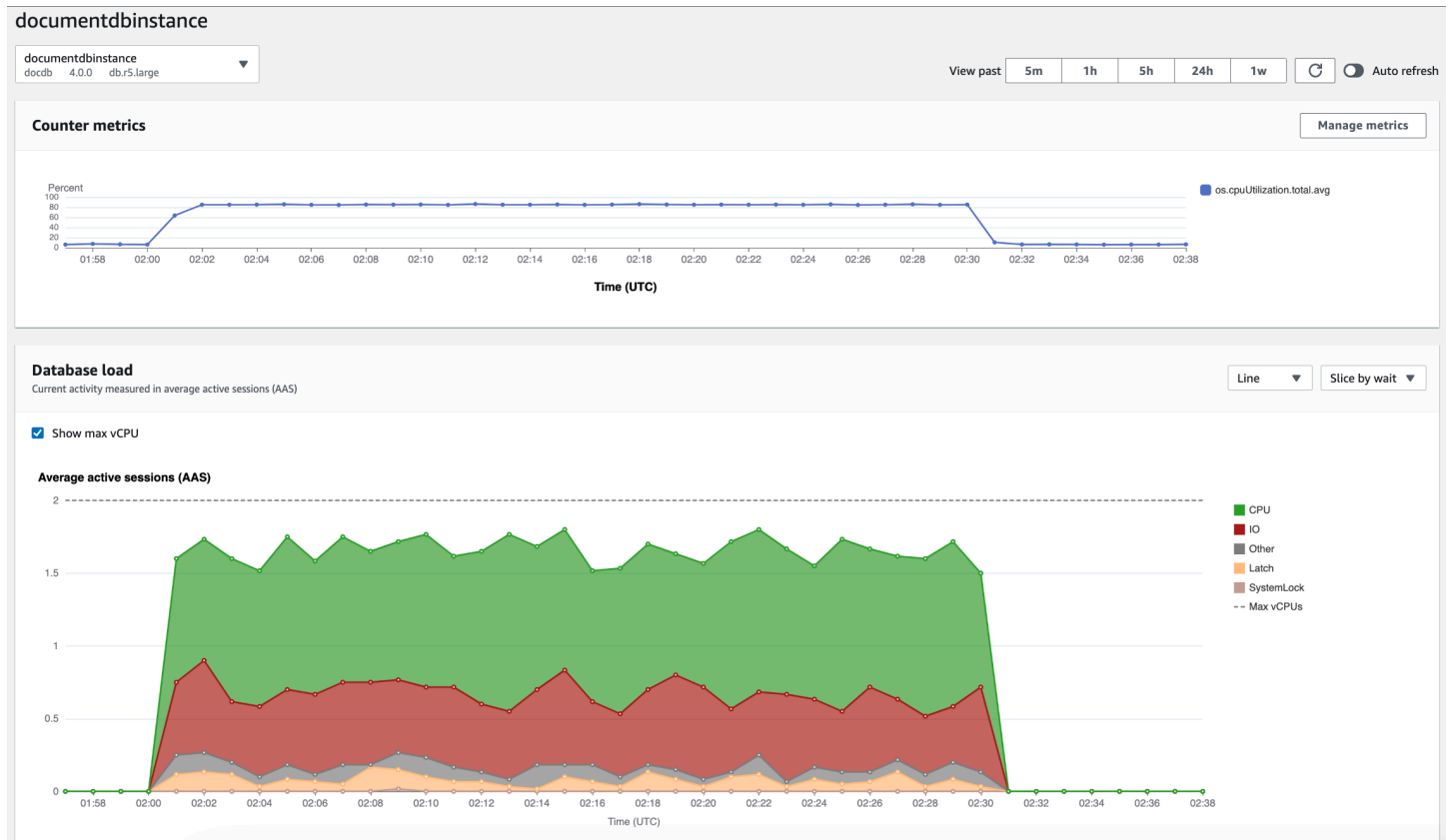
績效詳情儀表板包含資料庫效能資訊，可協助您分析效能問題並對其進行故障排除。在主儀表板頁面上，您可以查看有關資料庫負載 (DB load) 的信息。您可以按等待狀態或查詢等維度「切片」資料庫加載。

主題

- [績效詳情儀表板概觀](#)
- [開啟績效詳情儀表板](#)
- [依等待狀態分析資料庫負載](#)
- [「頂部查詢」頁籤概述](#)
- [放大資料庫載入圖](#)

績效詳情儀表板概觀

儀表板是與績效詳情進行互動的最簡單方式。下列範例顯示 Amazon DocumentDB 執行個體的儀表板。根據預設，績效詳情儀表板會顯示過去一小時內的資料。



儀表板分成下列部分：

1. 計數器量度 — 顯示特定效能計數器度量的資料。
2. 資料庫負載 — 顯示資料庫負載與資料庫執行個體容量的比較 (以最大 vCPU 線路表示)。
3. 頂端維度 — 顯示造成資料庫負載的頂端維度。這些維度包括waitsquerieshosts、databases、和applications。

主題

- [計數器指標圖表](#)
- [資料庫負載圖表](#)
- [最高維度表格](#)

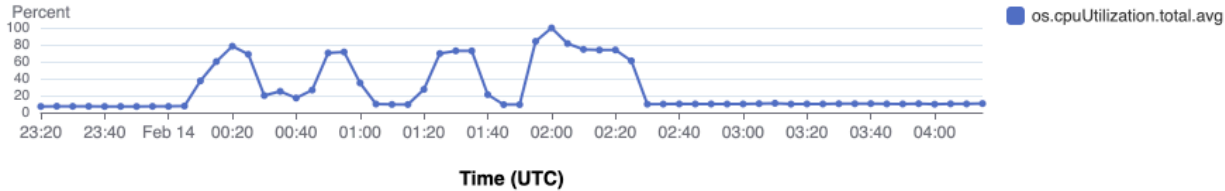
計數器指標圖表

您可以使用計數器指標來自訂績效詳情儀表板，以包含高達 10 個其它圖表。這些圖表顯示數十種作業系統指標的選擇。您可以將此資訊與資料庫負載相互關聯，以協助識別和分析效能問題。

Counter Metrics (計數器指標) 圖表顯示效能計數器的資料。

Counter metrics

Manage metrics



若要變更效能計數器，請選擇 [管理指標]。您可以選擇多個操作系統指標，如下面的螢幕截圖。若要查看任何指標的詳細資訊，請將游標移到指標名稱上。

Select metrics shown on the graph

✕

Check the metrics that you want to see on the Performance Insights dashboard.

OS metrics (4)

Clear all selections

▼ general

 numVCPU

▼ cpuUtilization

 idle system total user wait

▼ loadAverageMinute

 fifteen five one

▼ memory

 active buffers cached dirty free inactive

資料庫負載圖表

「資料庫負載」圖表顯示資料庫活動與執行處理容量的比較情形 (如「最大 vCPU」行所代表)。依預設，堆疊折線圖以每個時間單位的平均作用中工作階段數來表示資料庫負載。資料庫負載依等待狀態切割 (分組)。

Database load

Current activity measured in average active sessions (AAS)

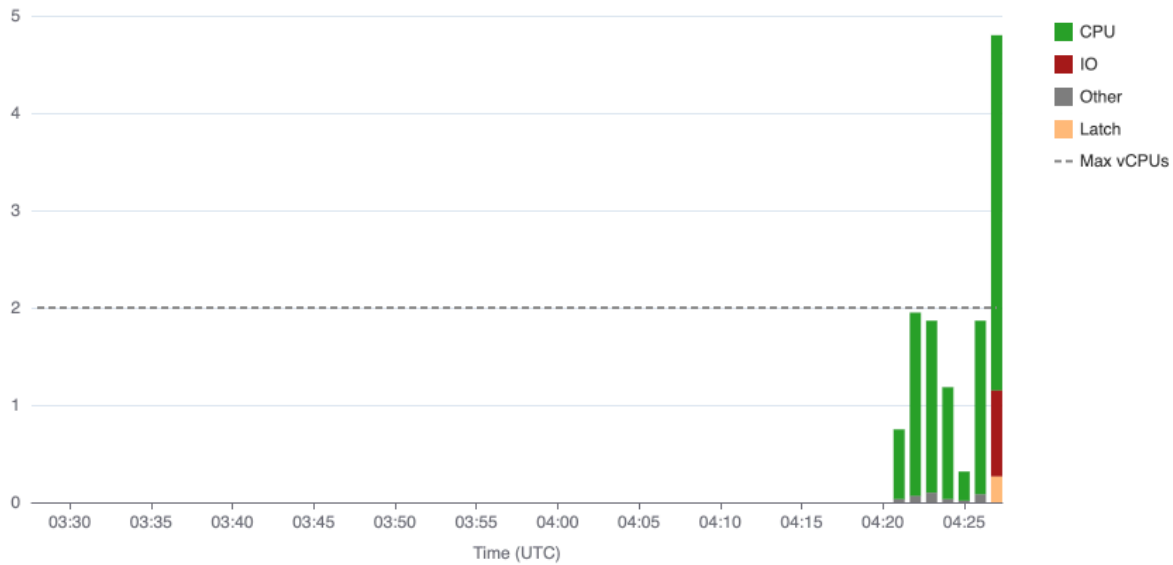
Bar



Slice by wait

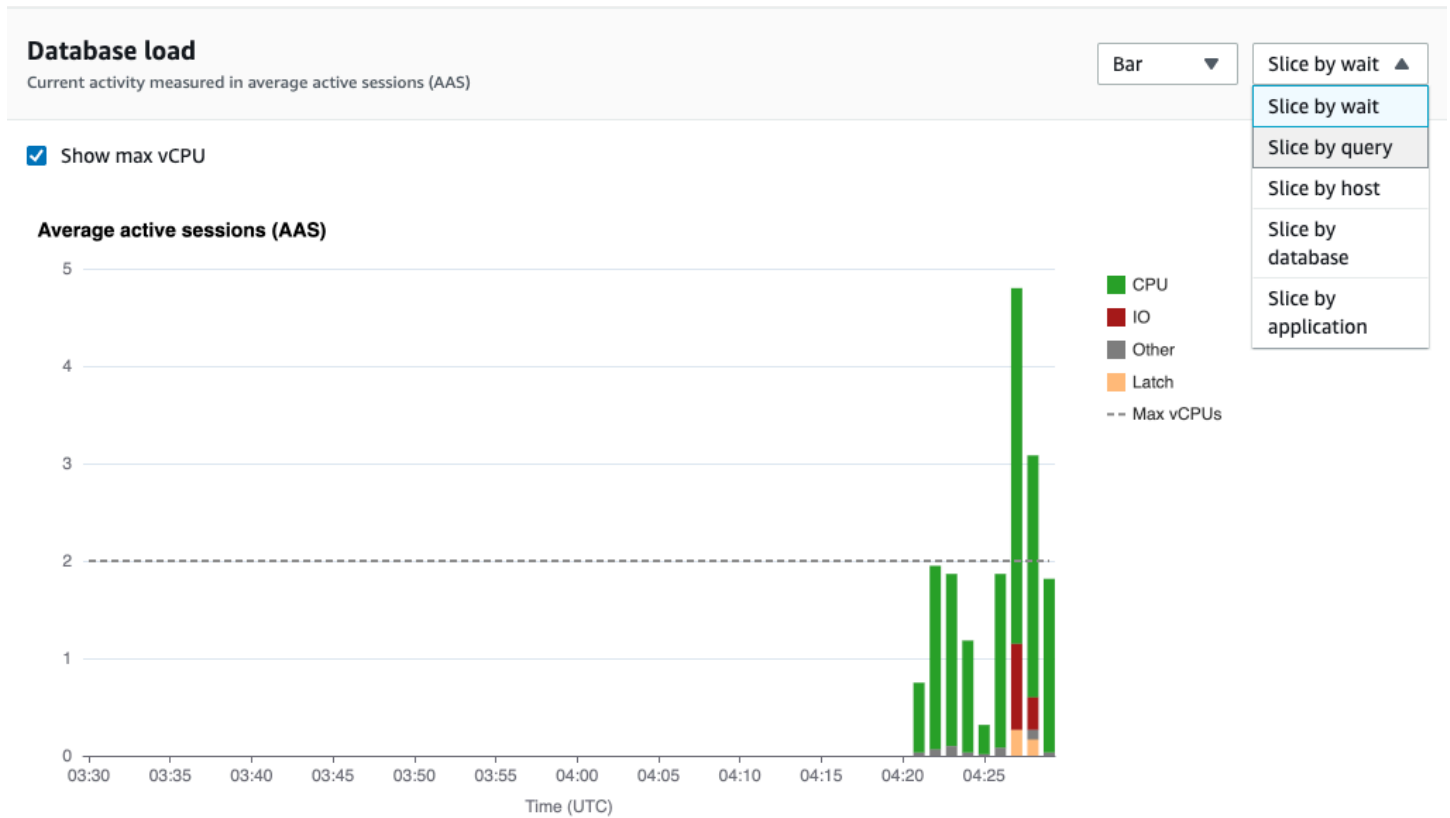
 Show max vCPU

Average active sessions (AAS)



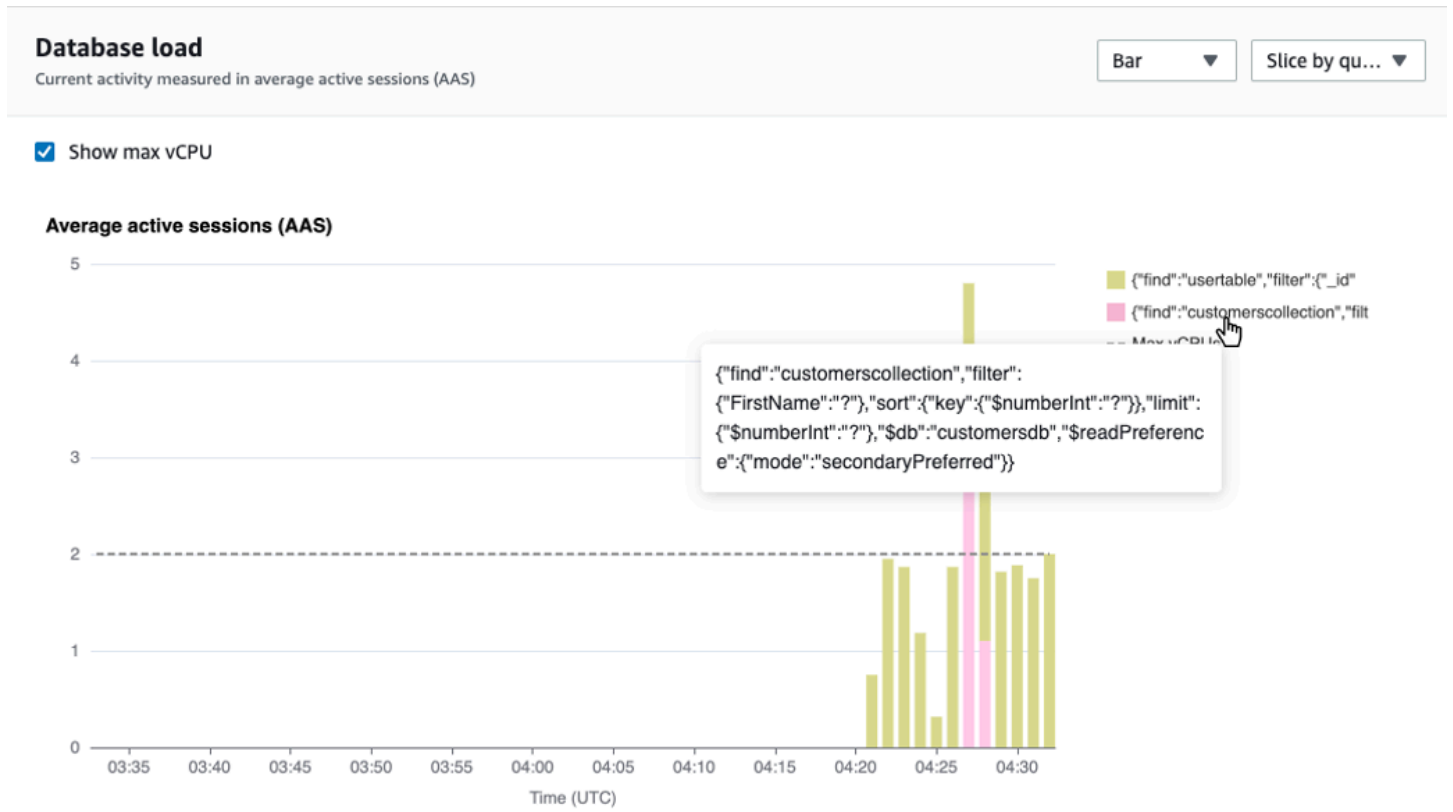
資料庫負載依維度配量

您可以選擇將負載顯示為作用中工作階段 (依任何支援維度分組)。下圖顯示 Amazon DocumentDB 執行個體的維度。

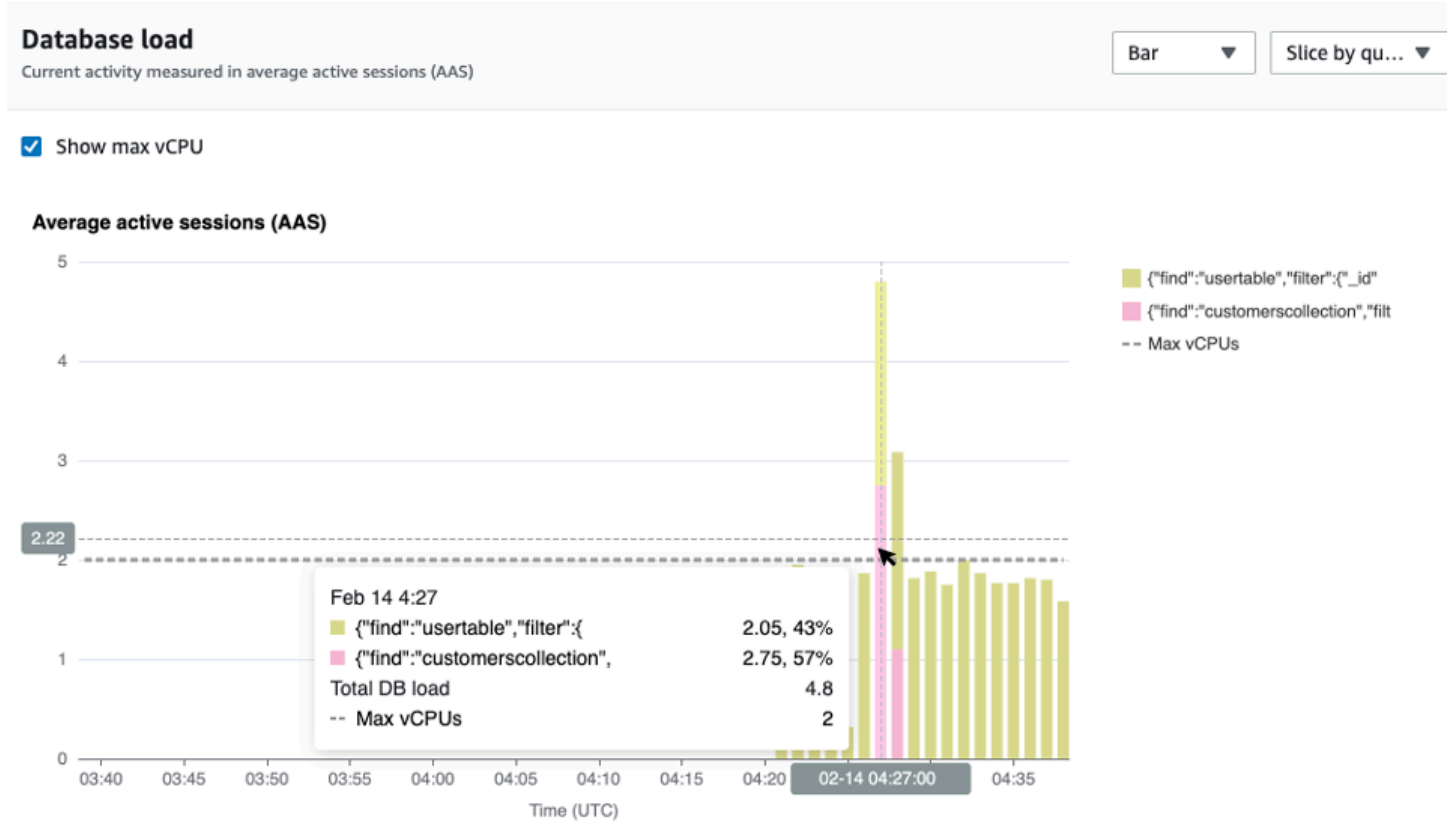


維度項目的資料庫負載詳細資訊

若要查看有關維度內資料庫負載項目的詳細資訊，請將游標移到項目名稱上。下列影像顯示查詢陳述式的詳細資料。



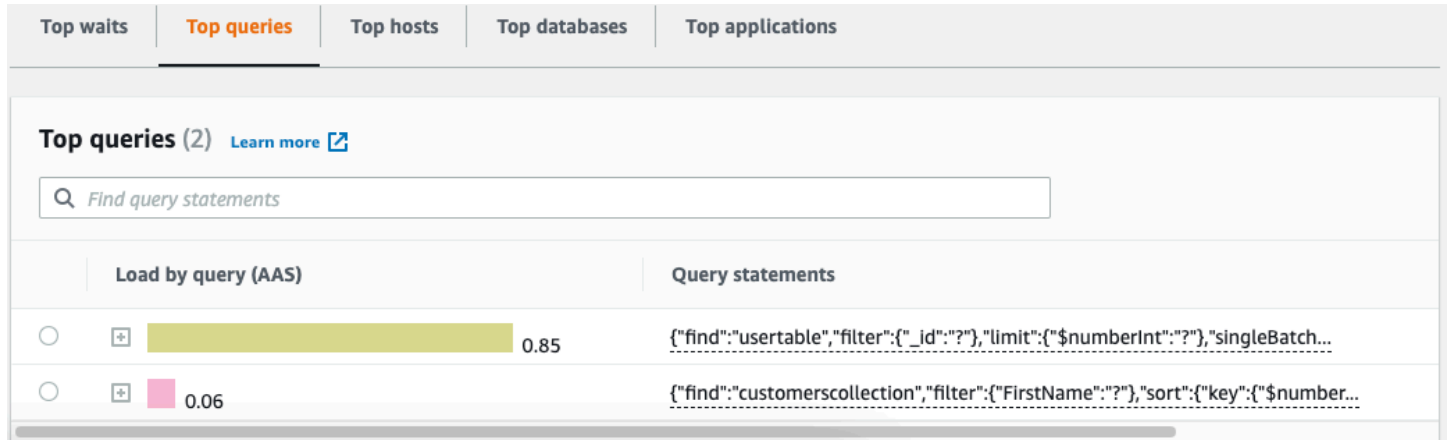
若要以圖例查看在所選時段內任何項目的詳細資訊，請將游標移到該項目上。



最高維度表格

頂部維度表按不同的維度切片 DB 負載。維度是資料庫負載各種特性的類別或「配量依據」。如果維度是查詢，「前幾個查詢」會顯示對資料庫負載貢獻最大的查詢陳述式。

選擇下列任一維度索引標籤：



下表提供每個標籤的簡短描述。

描述

竊聽器後端正在等待的事件
熱門查詢

描述執行的查詢陳述式最蘆銀欄戶端的主機 IP 和連接埠

描述
最高數據艙的資料庫名稱
數據應廣程序的應用程式名稱

若要瞭解如何使用 [常用查詢] 索引標籤分析查詢，請參閱 [〈「頂部查詢」頁籤概述](#)。

開啟績效詳情儀表板

若要在AWS管理主控台中檢視 [Performance Insights] 儀表板，請使用下列步驟：

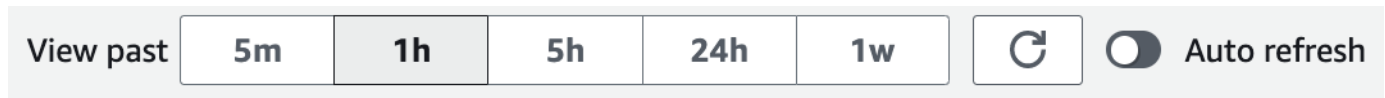
1. 開啟 Performance Insights 主控台，[網址為 https://console.aws.amazon.com/docdb/](https://console.aws.amazon.com/docdb/)。

- 選擇資料庫執行個體。此時會顯示該 Amazon DocumentDB 執行個體的 Performance Insights 儀表板。

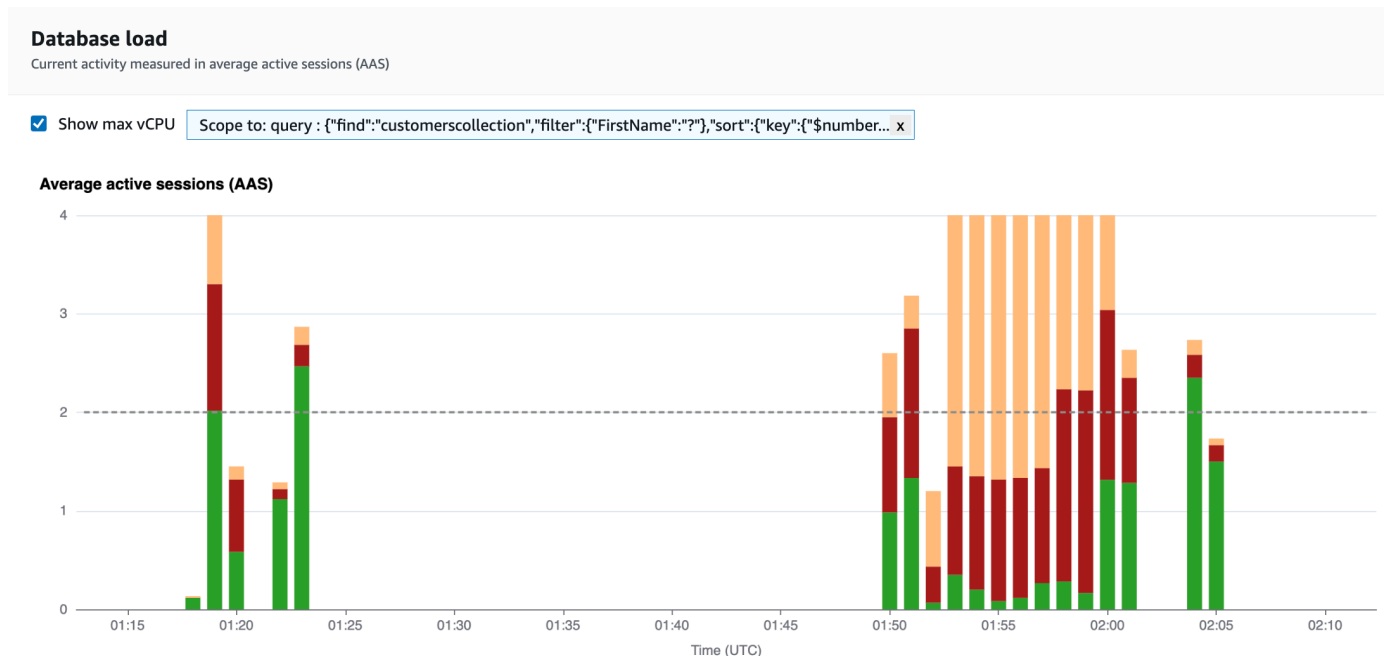
對於已啟用 Performance Insights 的 Amazon DocumentDB 執行個體，您也可以選擇執行個體清單中的工作階段項目，以存取儀表板。在 Current activity (目前活動) 中，Sessions (工作階段) 項目顯示了過去五分鐘內平均作用中工作階段的資料庫負載。負載以進度條圖形的方式顯示。當長條為空時，執行個體處於閒置狀態。隨著負載增加，進度條會填入藍色。當負載超過執行個體類別上的虛擬 CPU (vCPUs) 數目時，長條會變成紅色，表示潛在的瓶頸。

| Clusters (1) | | | | | | | | | |
|--------------------------|---------------------|------------------|----------------|-------------|-----------|--------|------------------|--|--|
| Filter Resources | | | | | | | | | |
| <input type="checkbox"/> | Cluster identifier | Role | Engine version | Region & AZ | Status | CPU | Current activity | | |
| <input type="checkbox"/> | documentdbinstance | Regional cluster | 4.0.0 | ap-south-1 | available | - | - | | |
| <input type="checkbox"/> | documentdbinstance | Primary instance | 4.0.0 | ap-south-1c | available | 84.99% | 5 Connections | | |
| <input type="checkbox"/> | documentdbinstance2 | Replica instance | 4.0.0 | ap-south-1b | available | 15.37% | 2 Connections | | |
| <input type="checkbox"/> | documentdbinstance3 | Replica instance | 4.0.0 | ap-south-1a | available | 14.84% | 2 Connections | | |

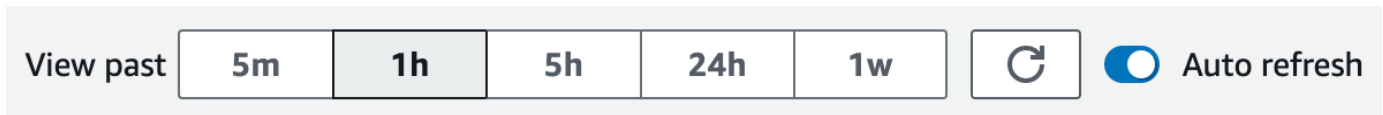
- (選用) 選取右上角中的按鈕，以選擇不同的時間間隔。例如，若要將間隔變更為 1 小時，請選取 1h。



在下面的屏幕截圖中，數據庫加載間隔為 1 小時。



- 若要自動重新整理資料，請啟用自動重新整理。



績效詳情儀表板會自動以新資料進行重新整理。重新整理速度取決於顯示的資料量：

- 5 分鐘的資料每 5 秒重新整理一次。
- 1 小時內的資料每分鐘重新整理一次。
- 5 小時內的資料每分鐘重新整理一次。
- 24 小時內的資料每 5 分鐘重新整理一次。
- 1 週內的資料每小時重新整理。

依等待狀態分析資料庫負載

如果資料庫負載 (DB 負載) 圖表顯示瓶頸，您可以找出負載來自何處。若要這麼做，請查看資料庫負載圖表下的最高負載項目表格。選擇特定項目 (例如查詢或應用程式)，以向下鑽研該項目並查看有關該項目的詳細資訊。

依等待和熱門查詢分組的資料庫負載通常可提供效能問題的最深入分析。根據等待分組的資料庫負載顯示該資料庫中是否有任何資源或正在發生的瓶頸。在此情況下，最上層載入項目表格的 [前幾個查詢] 索引標籤會顯示哪些查詢正在驅動該負載。

診斷效能問題的典型工作流程如下：

1. 檢閱資料庫負載圖表並查看是否有任何資料庫負載超越最高 CPU 線的情況。
2. 若有，請查看資料庫負載圖表，並找出哪一個或那幾個等待狀態是主因。
3. 藉由查看最上層載入項目表格上的「常用查詢」標籤對這些等待狀態最有貢獻的查詢，來識別造成負載的摘要查詢。您可以透過等待載入 (AAS) 資料行來識別這些項目。
4. 在 [前幾個查詢] 索引標籤中選擇其中一個摘要查詢，以展開它並查看其所組成的子查詢。

您也可以分別選取常用主機或常用應用程式，查看哪些主機或應用程式負載最多。應用程式名稱是在連接至 Amazon DocumentDB 執行個體的連接字串中指定。Unknown 表示未指定應用程式欄位。

例如，在下列儀表板中，CPU 會等待大部分資料庫負載的帳戶。選取「常用查詢」底下的常用查詢會將「資料庫負載」圖表的範圍集中在 select 查詢所貢獻的最多負載上。

Database load

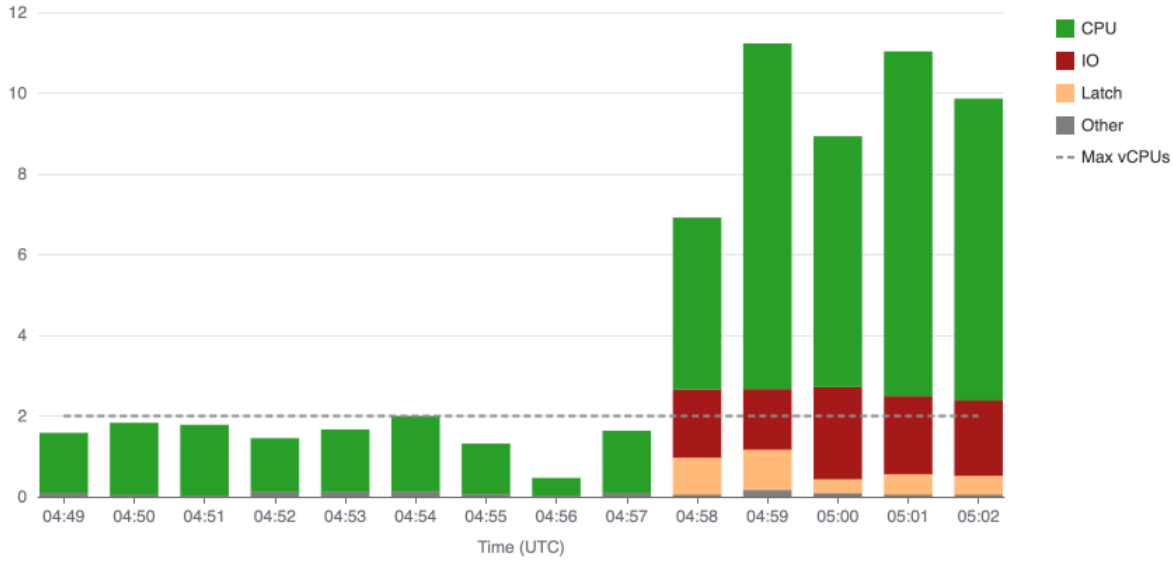
Current activity measured in average active sessions (AAS)

Bar ▼

Slice by wait ▼

Show max vCPU

Average active sessions (AAS)





- Top waits
- Top queries**
- Top hosts
- Top databases
- Top applications

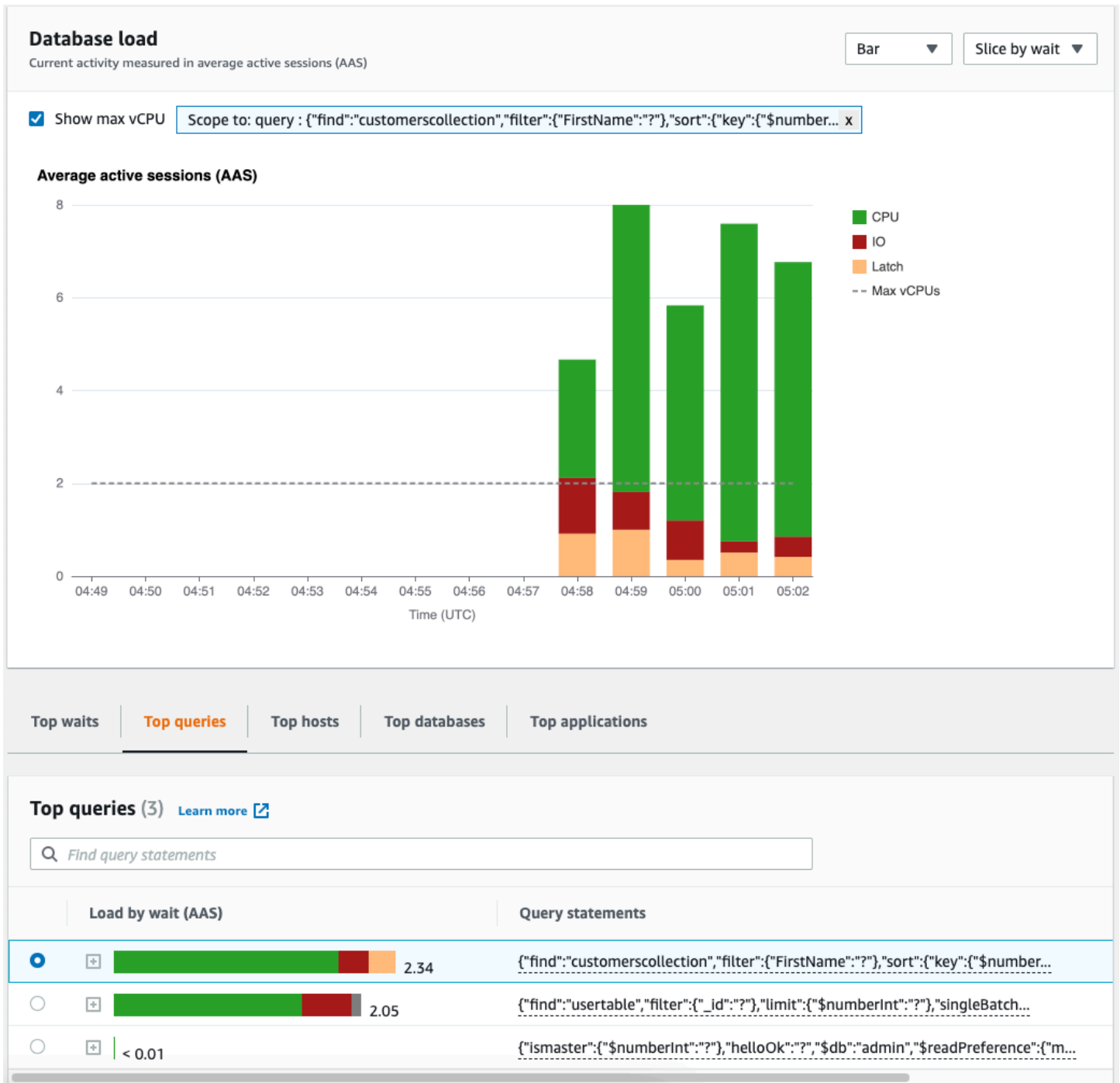
Top queries (3) [Learn more](#)

Find query statements

Load by wait (AAS)

Query statements

| | | | | |
|-----------------------|--------------------------|---|------|---|
| <input type="radio"/> | <input type="checkbox"/> |  | 2.34 | <code>{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":{"\$number...</code> |
| <input type="radio"/> | <input type="checkbox"/> |  | 2.05 | <code>{"find":"usertable","filter":{"_id":"?"},"limit":{"\$numberInt":"?"},"singleBatch...</code> |
| <input type="radio"/> | <input type="checkbox"/> | < 0.01 | | <code>{"ismaster":{"\$numberInt":"?"},"helloOk":"?","\$db":"admin","\$readPreference":{"m...</code> |



「頂部查詢」頁籤概述

根據預設，[頂端查詢] 索引標籤會顯示對資料庫負載貢獻最大的查詢。您可以分析查詢文字以協助調整查詢。

主題

- [查詢摘要](#)

- [依等待分組的負載 \(AAS\)](#)
- [檢視詳細的查詢資訊](#)
- [存取陳述式查詢文字](#)
- [檢視和下載陳述式查詢文字](#)

查詢摘要

查詢摘要為多個實際查詢的複合，這些查詢在結構上相似，但可能具有不同的常值。摘要中，問號會取代硬式編碼值。例如，查詢摘要可能如下所示：

```
{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":{"$numberInt":"?"}},"limit":{"$numberInt":"?"}}
```

此摘要可能包含下列子查詢：

```
{"find":"customerscollection","filter":{"FirstName":"Karrie"},"sort":{"key":{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
{"find":"customerscollection","filter":{"FirstName":"Met"},"sort":{"key":{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
{"find":"customerscollection","filter":{"FirstName":"Rashin"},"sort":{"key":{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
```

若要查看摘要中的常值查詢陳述式，請選取查詢，然後選擇加號 (+)。在下列螢幕擷取畫面中，所選查詢為摘要。

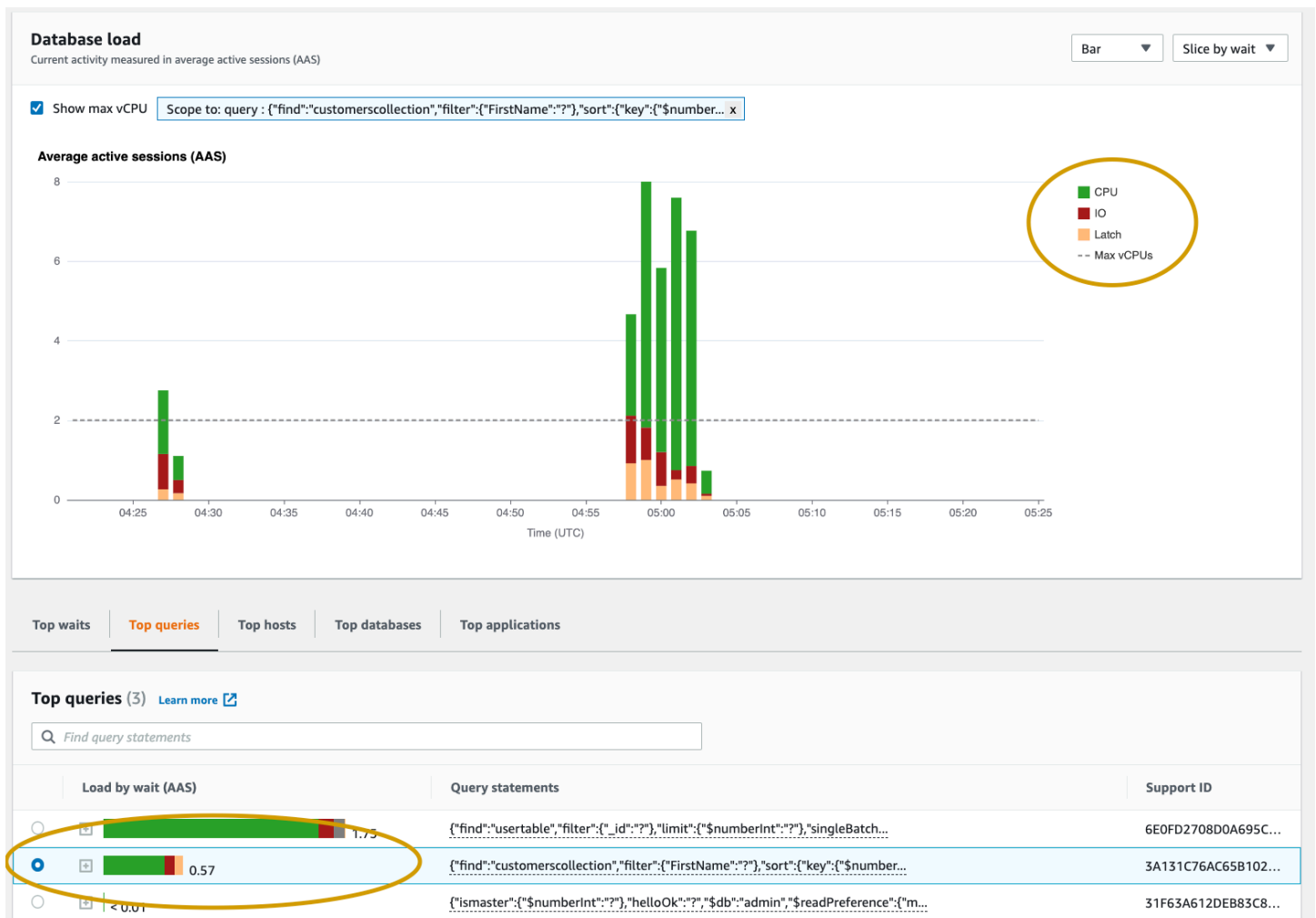
| Top waits | Top queries | Top hosts | Top databases | Top applications |
|--|-------------------------------|---|---------------|------------------|
| Top queries (3) Learn more | | | | |
| <input type="text" value="Find query statements"/> | | | | |
| | Load by wait (AAS) | Query statements | | |
| <input type="radio"/> | <input type="checkbox"/> 1.27 | <pre>{"find":"usertable","filter":{"_id":"?"},"limit":{"\$numberInt":"?"},"singleBatch...</pre> | | |
| <input type="radio"/> | <input type="checkbox"/> 0.41 | <pre>{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":{"\$number...</pre> | | |
| <input checked="" type="radio"/> | <input type="checkbox"/> 0.02 | <pre>{"find":"customerscollection","filter":{"FirstName":"Jesse"},"sort":{"key":{"\$nu...</pre> | | |
| <input type="radio"/> | <input type="checkbox"/> 0.02 | <pre>{"find":"customerscollection","filter":{"FirstName":"Jesse"},"sort":{"key":{"\$nu...</pre> | | |

Note

查詢摘要會將類似的查詢陳述式分組，但不會編輯敏感資訊。

依等待分組的負載 (AAS)

在「前幾個查詢」中，「等待載入 (AAS)」欄會說明與每個最上層載入項目相關聯的資料庫負載百分比。此資料欄會透過資料庫負載圖表中目前選取的任何群組來反映該項目的負載。例如，您可以依等待狀態將 DB load (資料庫負載) 圖表分組。在此情況下，根據等待列出資料庫負載列較大且分段，並以顏色為代碼來顯示該查詢所帶來的指定等待狀態程度。這也會顯示哪些等待狀態會影響選取的查詢。



檢視詳細的查詢資訊

在「前幾個查詢」表格中，您可以開啟摘要陳述式來檢視其資訊。資訊會顯示在底部窗格中。

Top waits | **Top queries** | Top hosts | Top databases | Top applications

Top queries (3) [Learn more](#)

Find query statements

| Load by wait (AAS) | Query statements | Support ID |
|--------------------|--|---------------------|
| 1.75 | { "find": "usertable", "filter": { "_id": "?" }, "limit": { "\$numberInt": "?" }, "singleBatch..." | 6E0FD2708D0A695C... |
| 0.57 | { "find": "customerscollection", "filter": { "FirstName": "?" }, "sort": { "key": { "\$number..." | 3A131C76AC65B102... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | 7C19C88DD78407E0... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | FBF2993E2172CF6G... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | 77449E3F829AC210... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | 01B0434C5D4F140D... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | D995AB7F6C835AE7... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | 613864818FDD36E2... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | 49537B8EA74BE915... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | 098E33A525332BBC... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | 792692547FD45F14... |
| 0.03 | { "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "\$nu... | 367B900BA7E20C39... |
| < 0.01 | { "ismaster": { "\$numberInt": "?" }, "helloOk": "?", "\$db": "admin", "\$readPreference": { "m... | 31F63A612DEB83C8... |

Query information

```
{ "find": "customerscollection", "filter": { "FirstName": "Jesse" }, "sort": { "key": { "$numberInt": "1" }, "limit": { "$numberInt": "3" }, "lsid": { "id": { "$binary": { "base64": "DG/4c0FlRxywzmItINb+MA==", "subType": "04" } } }, "$db": "customersdb", "$readPreference": { "mode": "secondaryPreferred" } }
```

Query ID: pi-563169974 ([Support query ID](#)) Digest ID: pi-563169974 ([Support Digest ID](#))

[Copy](#) [Download](#)

下列類型的識別碼 (ID) 與查詢陳述式相關聯：

1. S@@ upport 查詢 ID — 查詢 ID 的雜湊值。當您使用「Support AWS 援」時，此值僅用於參考查詢 ID。AWSSupport 無法存取您實際的查詢 ID 和查詢文字。
2. S@@ upport 摘要 ID — 摘要 ID 的雜湊值。此值只是在您與 AWS Support 互動時用來參考摘要 ID。AWSSupport 無法存取您的實際摘要 ID 和查詢文字。

存取陳述式查詢文字

根據預設，「前幾個查詢」表格中的每一列都會針對每個查詢陳述式顯示 500 個位元組的查詢文字。當摘要陳述式超過 500 位元組時，您可以在「Performance Insights」儀表板中開啟陳述式，以檢視更多文字。在這種情況下，顯示的查詢長度上限為 1 KB。如果您檢視完整的查詢陳述式，也可以選擇 [下載]。

檢視和下載陳述式查詢文字

在「Performance Insights」儀表中，您可以檢視或下載查詢文字。

若要在 [Performance Insights] 儀表中檢視更多查詢文字

1. 在以下位置打開 Amazon DocumentDB 控制台：<https://console.aws.amazon.com/docdb/>
2. 在導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。顯示該資料庫執行個體的績效詳情儀表板。

文字大於 500 位元組的查詢陳述式看起來如下圖所示：

| Load by wait (AAS) | Query statements | Support ID |
|--------------------|--|---------------------|
| 1.75 | <code>{"find":"usertable","filter":{"_id":"?"},"limit":{"\$numberint":"?"},"singleBatch...</code></code> | 6E0FD2708D0A695C... |
| 0.57 | <code>{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":{"\$number...</code></code> | 3A131C76AC65B102... |
| 0.03 | <code>{"find":"customerscollection","filter":{"FirstName":"Jesse"},"sort":{"key":{"\$nu...</code></code> | 7C19C8DD78407E0... |
| 0.03 | <code>{"find":"customerscollection","filter":{"FirstName":"Jesse"},"sort":{"key":{"\$nu...</code></code> | FBF2993E2172CFC6... |

4. 檢查查詢資訊區段以檢視更多查詢文字。

```
{"find":"customerscollection","filter":{"FirstName":"Jesse"},"sort":{"key":{"$numberInt":"1"}},"limit":{"$numberInt":"3"},"lsid":{"id":{"$binary":<br>{"base64":"DG/4c0FLRxywzmtINb+MA==","subType":"04"}}},"$db":"customersdb","$readPreference":{"mode":"secondaryPreferred"}}
```

Query ID: pi-563169974 (Support query ID) Digest ID: pi-563169974 (Support Digest ID)

Copy Download

每個完整查詢陳述式的「Performance Insights」儀表板最多可顯示 1 KB。

Note

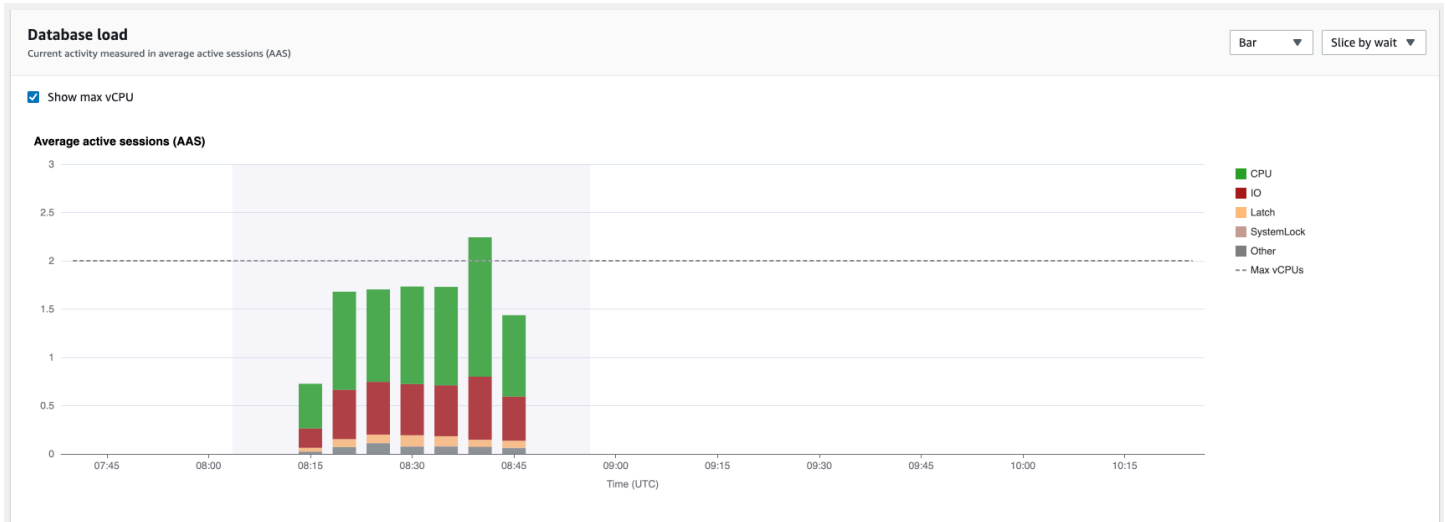
若要複製或下載查詢陳述式，請停用任何快顯封鎖程式。

放大資料庫載入圖

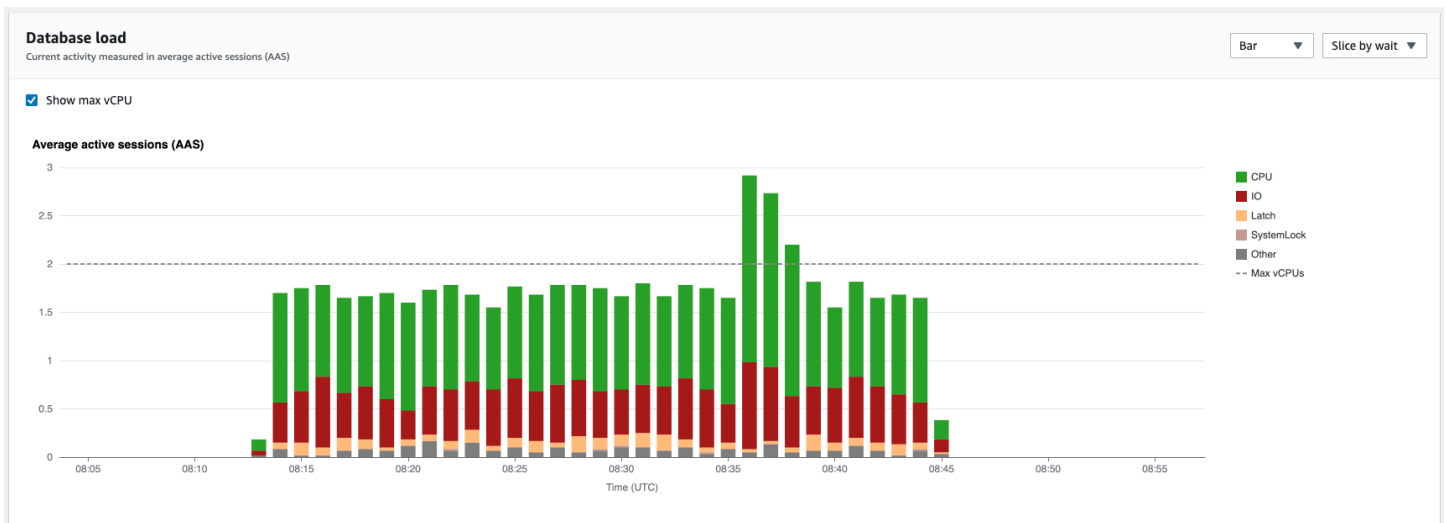
您可以使用績效詳情使用者界面的其他功能來協助並分析效能資料。

長按拖曳來放大

在績效詳情界面中，您可以選擇負載圖表的一小部分，並放大詳細資訊。



若要放大小部分的負載表格，選擇開始時間並拖曳到您想要的時間範圍結束時間。您這麼做的話，所選區域將有反白顯示。當您放開滑鼠時，載入圖表會放大選取的區域，並重新計算「前幾個項目」表格。



[Top waits](#) | [Top queries](#) | [Top hosts](#) | [Top databases](#) | [Top applications](#)

Top waits (5)

| | Load by wait (AAS) | Wait |
|-----------------------|--|------|
| <input type="radio"/> |  0.63 | CPU |
| <input type="radio"/> |  0.42 | IO |

使用績效詳情 API 來擷取指標

在啟用績效詳情時，API 會提供對執行個體效能的可見性。Amazon CloudWatch Logs 為服務的付費監控指標提供授權來源。AWS

績效詳情提供以平均作用中工作階段 (AAS) 評估的資料庫負載特定網域檢視。此指標在 API 消費者看來是二維時間序列資料集。資料的時間維度提供查詢的時間範圍內各時間點的資料庫負載資料。每個時間點會根據請求的維度來分解整體負載，例如 Query、Wait-state、Application、或者 Host，在該時間點所測得。

Amazon DocumentDB Performance Insights 可監控您的 Amazon DocumentDB 資料庫執行個體，以便您分析資料庫效能並進行疑難排解。檢視績效詳情資料的一個方法就是使用 AWS Management Console。績效詳情也提供公有 API，讓您可以查詢自己的資料。您可以使用 API 執行下列動作：

- 將資料卸載至資料庫
- 將績效詳情資料新增至現有監控儀表板
- 建置監控工具

若要使用 Performance Insights API，請在其中一個 Amazon DocumentDB 執行個體上啟用 Performance Insights 見。如需啟用績效詳情的相關資訊，請參閱 [啟用和停用績效詳情](#)。如需績效詳情 API 的相關詳細資訊，請參閱 [績效詳情 API 參考](#)。

績效詳情 API 提供下列操作。

| 績效詳情動作 | AWS CLI 命令 | 描述 |
|--|--|--|
| DescribeDimensionKeys | aws pi describe-dimension-keys | 針對特定時段，擷取其指標的前 N 個維度金鑰。 |
| GetDimensionKeyDetails | aws pi get-dimension-key-details | 擷取資料庫執行個體或資料來源之指定維度群組的屬性。例如，如果您指定查詢 ID，並且維度詳細資訊可用，則 <code>GetDimensionKeyDetails</code> 會擷取與此 ID <code>db.query.statement</code> 相關聯之維度的全文。這項作業很有用，因為 <code>GetResourceMetrics</code> 且 <code>DescribeDimensionKeys</code> 不支援擷取大型查詢陳述式文字。 |

| 績效詳情動作 | AWS CLI 命令 | 描述 |
|---|---|--|
| GetResourceMetadata | aws pi get-resource-metadata | 檢索不同功能的中繼資料。例如，中繼資料可能指出特定資料庫執行個體上某項功能已開啟或關閉。 |
| GetResourceMetrics | aws pi get-resource-metrics | 擷取一組資料來源某個時段的績效詳情指標。您可以提供特定維度群組和維度，以及為每個群組提供彙總和篩選條件。 |
| ListAvailableResourceDimensions | aws pi list-available-resource-dimensions | 檢索指定執行個體上每個指定指標類型可查詢的維度。 |
| ListAvailableResourceMetrics | aws pi list-available-resource-metrics | 檢索可為指定資料庫執行個體查詢的指定指標類型中所有可用的指標。 |

主題

- [AWS CLI 績效詳情的](#)
- [擷取時間序列指標](#)
- [績效詳情的 AWS CLI 範例](#)

AWS CLI 績效詳情的

您可以使用 AWS CLI 檢視績效詳情資料。您可以透過在命令列輸入以下內容來檢視績效詳情 AWS CLI 命令的說明。

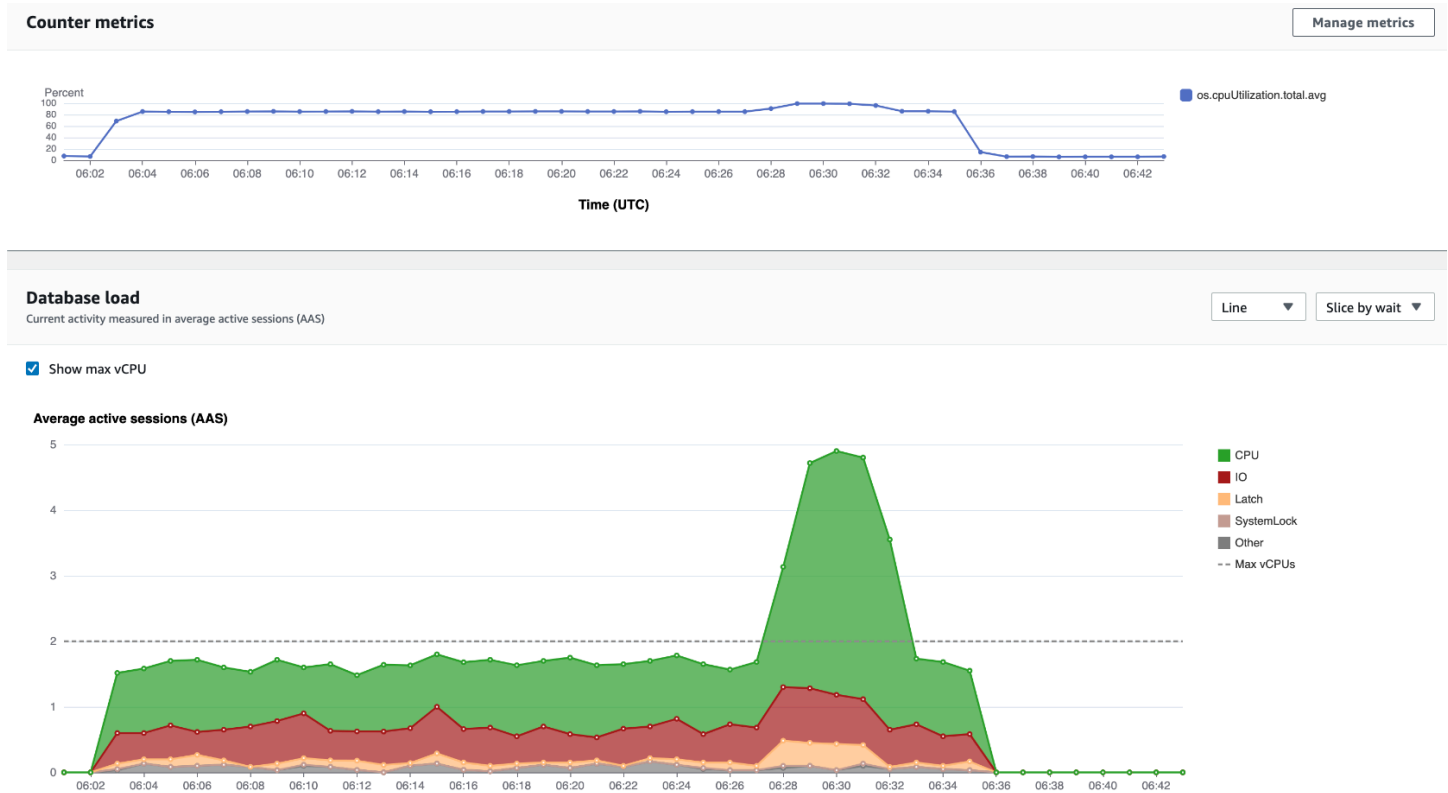
```
aws pi help
```

如果您未安裝 AWS CLI，請參閱《AWS CLI 使用者指南》中的[安裝 AWS 命令列介面](#)來取得安裝資訊。

擷取時間序列指標

GetResourceMetrics 操作會從績效詳情資料中擷取一或多個時間序列指標。GetResourceMetrics 需要指標和時間間隔，並傳回含資料點清單的回應。

例如，AWS Management Console 會將 GetResourceMetrics 用於填入 Counter Metrics (計數器指標) 圖表和 Database Load (資料庫負載) 圖表，如下圖所示。



GetResourceMetrics 傳回的所有指標，除 `db.load` 之外，皆為標準的時間序列指標。此指標會顯示在 Database Load (資料庫負載) 圖表中。`db.load` 指標與其他時間序列指標不同，因為您可以將它分為名為維度的子元件。在先前的影像中，`db.load` 已被細分，分組依據為組成 `db.load` 的等待狀態。

Note

GetResourceMetrics 也可以傳回 `db.sampleload` 指標，但 `db.load` 指標適用於大部分情況。

如需 GetResourceMetrics 所傳回指標的相關資訊，請參閱[計數器指標的 Performance Insights](#)。

這些指標支援下列計算：

- 平均值 – 指標在一段時間內的平均值。將 `.avg` 附加至指標名稱。
- 最小值 – 指標在一段時間內的最小值。將 `.min` 附加至指標名稱。
- 最大值 – 指標在一段時間內的最大值。將 `.max` 附加至指標名稱。
- 總和 – 指標值在一段時間內的總和。將 `.sum` 附加至指標名稱。
- 取樣計數 – 在一段時間內收集指標的次數。將 `.sample_count` 附加至指標名稱。

例如，假設收集指標的時間為 300 秒 (5 分鐘)，且每分鐘收集一次指標。每分鐘的值為 1、2、3、4 和 5。在此情況下，會傳回下列計算：

- 平均值 – 3
- 最小值 – 1
- 最大值 – 5
- 總和 – 15
- 取樣計數 – 5

如需使用 `get-resource-metrics` AWS CLI 命令的相關資訊，請參閱 [get-resource-metrics](#)。

對於 `--metric-queries` 選項，請指定您要取得結果的一或多個查詢。每個查詢的組成為必要的 `Metric` 和選用的 `GroupBy` 及 `Filter` 參數。以下是 `--metric-queries` 選項規格的範例。

```
{
  "Metric": "string",
  "GroupBy": {
    "Group": "string",
    "Dimensions": ["string", ...],
    "Limit": integer
  },
  "Filter": {"string": "string"
  ...}
```

績效詳情的 AWS CLI 範例

下列範例示範如何使用績效詳情的 AWS CLI。

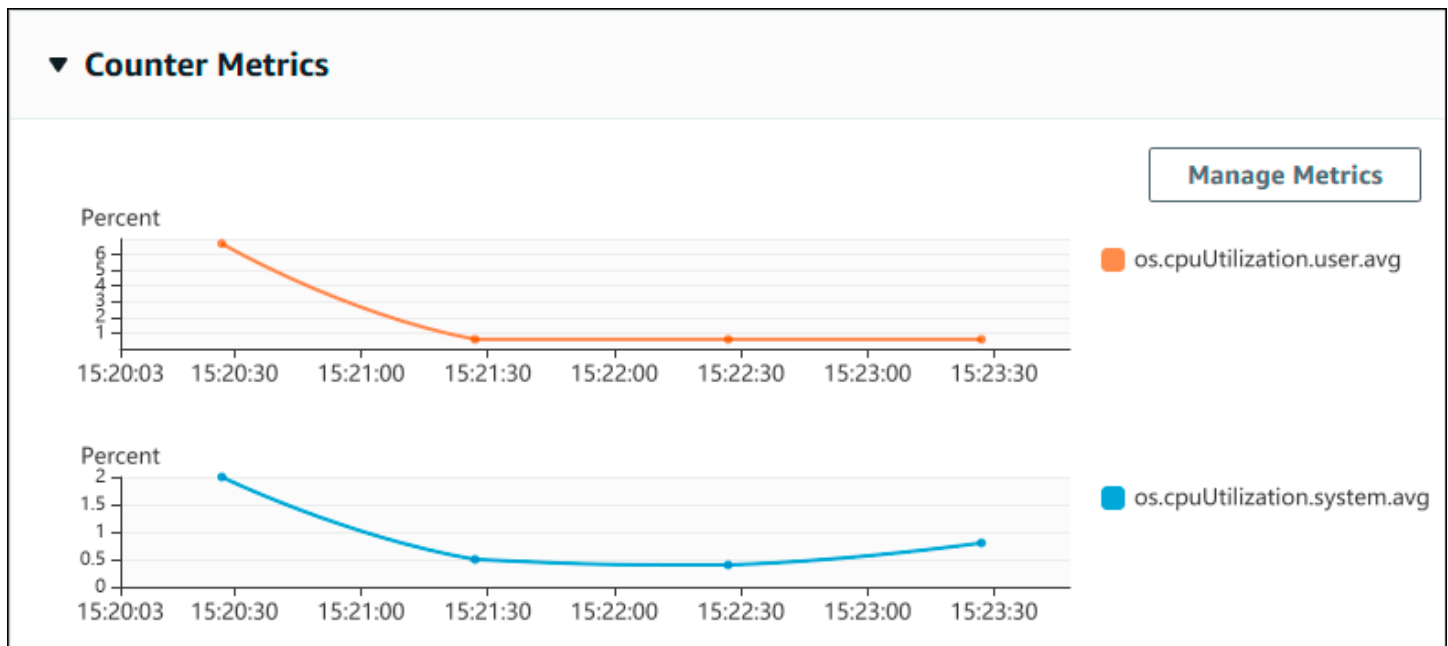
主題

- [擷取計數器指標](#)

- [擷取前幾個等待狀態的資料庫負載平均值](#)
- [擷取最上層查詢的資料庫負載平均值](#)
- [擷取由查詢篩選的資料庫負載平均值](#)

擷取計數器指標

下列螢幕擷取畫面顯示 AWS Management Console 中的兩個計數器指標圖表。



下凸顯是如何蒐集 AWS Management Console 用來產生兩個計數器指標圖表的相同資料。

針對 Linux、macOS 或 Unix：

```
aws pi get-resource-metrics \
  --service-type DOCDB \
  --identifier db-ID \
  --start-time 2022-03-13T8:00:00Z \
  --end-time 2022-03-13T9:00:00Z \
  --period-in-seconds 60 \
  --metric-queries '[{"Metric": "os.cpuUtilization.user.avg" },
                    {"Metric": "os.cpuUtilization.idle.avg"}]'
```

針對 Windows：

```
aws pi get-resource-metrics ^
  --service-type DOCDB ^
```

```
--identifier db-ID ^
--start-time 2022-03-13T8:00:00Z ^
--end-time 2022-03-13T9:00:00Z ^
--period-in-seconds 60 ^
--metric-queries '[{"Metric": "os.cpuUtilization.user.avg" },
                  {"Metric": "os.cpuUtilization.idle.avg"}]'
```

您也可以透過指定 `--metrics-query` 選項的檔案來提高命令的可讀性。以下範例會將名為 `query.json` 的檔案用於此選項。此檔案的內容如下。

```
[
  {
    "Metric": "os.cpuUtilization.user.avg"
  },
  {
    "Metric": "os.cpuUtilization.idle.avg"
  }
]
```

執行下列命令來使用檔案。

針對 Linux、macOS 或 Unix：

```
aws pi get-resource-metrics \
  --service-type DOCDB \
  --identifier db-ID \
  --start-time 2022-03-13T8:00:00Z \
  --end-time 2022-03-13T9:00:00Z \
  --period-in-seconds 60 \
  --metric-queries file://query.json
```

針對 Windows：

```
aws pi get-resource-metrics ^
  --service-type DOCDB ^
  --identifier db-ID ^
  --start-time 2022-03-13T8:00:00Z ^
  --end-time 2022-03-13T9:00:00Z ^
  --period-in-seconds 60 ^
  --metric-queries file://query.json
```

先前的範例會為選項指定下列值：

- `--service-type-DOCDB` 對於 Amazon DocumentDB
- `--identifier-` 資料執行個體的資源 ID
- `--start-time` 和 `--end-time` – 要查詢期間的 ISO 8601 DateTime 值，支援多種格式

它會查詢一小時的時間範圍：

- `--period-in-seconds-60` 適用於每分鐘的查詢
- `--metric-queries-` 兩個查詢的陣列，一個指標剛好一個查詢。

此指標名稱會使用點將指標分類在實用的類別，其中最後一個元素則做為函數。在此範例中，此函數是每個查詢的 `avg`。與亞馬遜一樣 CloudWatch，支援的功能是 `min` `max` `total`，，和 `avg`。

回應看起來類似以下的內容。

```
{
  "AlignedStartTime": "2022-03-13T08:00:00+00:00",
  "AlignedEndTime": "2022-03-13T09:00:00+00:00",
  "Identifier": "db-NQF3TTMFQ3GT0KIMJ0DMC3KQQ4",
  "MetricList": [
    {
      "Key": {
        "Metric": "os.cpuUtilization.user.avg"
      },
      "DataPoints": [
        {
          "Timestamp": "2022-03-13T08:01:00+00:00", //Minute1
          "Value": 3.6
        },
        {
          "Timestamp": "2022-03-13T08:02:00+00:00", //Minute2
          "Value": 2.6
        },
        //.... 60 datapoints for the os.cpuUtilization.user.avg metric
      ]
    },
    {
      "Key": {
        "Metric": "os.cpuUtilization.idle.avg"
      },
      "DataPoints": [
        {
          "Timestamp": "2022-03-13T08:01:00+00:00",
```

```

        "Value": 92.7
      },
      {
        "Timestamp": "2022-03-13T08:02:00+00:00",
        "Value": 93.7
      },
      //.... 60 datapoints for the os.cpuUtilization.user.avg metric
    ]
  }
] //end of MetricList
} //end of response

```

回應具有 Identifier、AlignedStartTime 和 AlignedEndTime。--period-in-seconds 值為 60，開始和結束時間皆一致使用分鐘。如果 --period-in-seconds 是 3600，開始和結束時間則會一致使用小時。

回應中的 MetricList 擁有許多項目，每個都包含 Key 和 DataPoints 項目。每個 DataPoint 都有 Timestamp 和 Value。每個 Datapoints 清單有 60 個資料點，因為查詢是適用於一小時中的每分鐘資料，內含 Timestamp1/Minute1、Timestamp2/Minute2 等，最多可達 Timestamp60/Minute60。

因為此查詢是適用於兩個不同的計數器指標，回應 MetricList 中會有兩個元素。

擷取前幾個等待狀態的資料庫負載平均值

系列範例是 AWS Management Console 用來產生堆疊區域折線圖的相同查詢。此範例會根據前七個等待狀態擷取負載劃分的最後一小時。db.load.avg 此命令與 [擷取計數器指標](#) 中的命令相同。然而，查詢 query.json 檔案有以下內容。

```

[
  {
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.wait_state", "Limit": 7 }
  }
]

```

執行下列命令。

針對 Linux、macOS 或 Unix：

```
aws pi get-resource-metrics \
```

```
--service-type DOCDB \
--identifier db-ID \
--start-time 2022-03-13T8:00:00Z \
--end-time 2022-03-13T9:00:00Z \
--period-in-seconds 60 \
--metric-queries file://query.json
```

針對 Windows :

```
aws pi get-resource-metrics ^
--service-type DOCDB ^
--identifier db-ID ^
--start-time 2022-03-13T8:00:00Z ^
--end-time 2022-03-13T9:00:00Z ^
--period-in-seconds 60 ^
--metric-queries file://query.json
```

此範例會指定前七個db.load.avg等待狀態GroupBy的測量結果和 a。如需此範例有效值的詳細資訊，請參閱 Performance Insights API 參考[DimensionGroup](#)中的。

回應看起來類似以下的內容。

```
{
  "AlignedStartTime": "2022-04-04T06:00:00+00:00",
  "AlignedEndTime": "2022-04-04T06:15:00+00:00",
  "Identifier": "db-NQF3TTMFQ3GTOKIMJ0DMC3KQQ4",
  "MetricList": [
    //A list of key/datapoints
    {
      "Key": {
        //A Metric with no dimensions. This is the total db.load.avg
        "Metric": "db.load.avg"
      },
      "DataPoints": [
        //Each list of datapoints has the same timestamps and same number of
items
        {
          "Timestamp": "2022-04-04T06:01:00+00:00",//Minute1
          "Value": 0.0
        },
        {
          "Timestamp": "2022-04-04T06:02:00+00:00",//Minute2
          "Value": 0.0
        }
      ]
    }
  ]
}
```

```

    },
    //... 60 datapoints for the total db.load.avg key
  ]
},
{
  "Key": {
    //Another key. This is db.load.avg broken down by CPU
    "Metric": "db.load.avg",
    "Dimensions": {
      "db.wait_state.name": "CPU"
    }
  },
  "DataPoints": [
    {
      "Timestamp": "2022-04-04T06:01:00+00:00", //Minute1
      "Value": 0.0
    },
    {
      "Timestamp": "2022-04-04T06:02:00+00:00", //Minute2
      "Value": 0.0
    },
    //... 60 datapoints for the CPU key
  ]
}, //... In total we have 3 key/datapoints entries, 1) total, 2-3) Top Wait
States
] //end of MetricList
} //end of response

```

在此回應中，中有三個項目MetricList。總共有一個項目db.load.avg，根據前三個等待狀態之一，db.load.avg分別有三個項目分別為三個項目。由於有分組維度 (與第一個範例不同)，因此量度的每個群組都必須有一個索引鍵。每個指標不能只有一個索引鍵，如同基本計數器指標使用案例。

擷取最上層查詢的資料庫負載平均值

下列範例會db.wait_state依前 10 名查詢陳述式分組。查詢陳述式有兩個不同的群組：

- db.query— 完整的查詢陳述式，例如 {"find":"customers","filter":{"FirstName":"Jesse"},"sort":{"key":{"\$numberInt":"1"}}}
- db.query_tokenized— 標記化的查詢陳述式，例如 {"find":"customers","filter":{"FirstName":"?"},"sort":{"key":{"\$numberInt":"?"}},"limit":{"\$numberInt":"?"}}

分析資料庫效能時，將只有參數不同的查詢陳述式視為一個邏輯項目會很有用。因此，您可以在查詢時使用 `db.query_tokenized`。但是，特別是當您感興趣時 `explain()`，有時檢查帶有參數的完整查詢語句更有用。在標記化查詢和完整查詢之間存在父子關係，多個完整查詢（子項）分組在同一個標記化查詢（父項）下。

此範例中的命令與 [擷取前幾個等待狀態的資料庫負載平均值](#) 中的命令類似。然而，查詢 `query.json` 檔案有以下內容。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.query_tokenized", "Limit": 10 }
  }
]
```

以下範例使用 `db.query_tokenized`。

針對 Linux、macOS 或 Unix：

```
aws pi get-resource-metrics \
  --service-type DOCDB \
  --identifier db-ID \
  --start-time 2022-03-13T8:00:00Z \
  --end-time 2022-03-13T9:00:00Z \
  --period-in-seconds 3600 \
  --metric-queries file://query.json
```

針對 Windows：

```
aws pi get-resource-metrics ^
  --service-type DOCDB ^
  --identifier db-ID ^
  --start-time 2022-03-13T8:00:00Z ^
  --end-time 2022-03-13T9:00:00Z ^
  --period-in-seconds 3600 ^
  --metric-queries file://query.json
```

這個範例會查詢超過 1 小時，且有一分鐘的時間 `period-in-seconds`。

此範例會指定前七個 `db.load.avg` 等待狀態 `GroupBy` 的測量結果和 `a`。如需此範例有效值的詳細資訊，請參閱 Performance Insights API 參考 [DimensionGroup](#) 中的。

回應看起來類似以下的內容。

```
{
  "AlignedStartTime": "2022-04-04T06:00:00+00:00",
  "AlignedEndTime": "2022-04-04T06:15:00+00:00",
  "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
  "MetricList": [
    { //A list of key/datapoints
      "Key": {
        "Metric": "db.load.avg"
      },
      "DataPoints": [
        //... 60 datapoints for the total db.load.avg key
      ]
    },
    {
      "Key": { //Next key are the top tokenized queries
        "Metric": "db.load.avg",
        "Dimensions": {
          "db.query_tokenized.db_id": "pi-1064184600",
          "db.query_tokenized.id": "77DE8364594EXAMPLE",
          "db.query_tokenized.statement": "{\"find\":\"customers\",\"filter\":"
          + "{\"FirstName\":\"?\"},\"sort\":{\"key\":{\"numberInt\":\"?\"}},\"limit\"":
          + "{\"numberInt\":\"?\"},\"$db\":\"myDB\",\"$readPreference\":{\"mode\":\"primary\"}}"}
        }
      },
      "DataPoints": [
        //... 60 datapoints
      ]
    },
    // In total 11 entries, 10 Keys of top tokenized queries, 1 total key
  ] //End of MetricList
} //End of response
```

此回應中有 11 個項目 MetricList (總共 1 個，10 個頂部標記化查詢)，每個項目每小時有 24 個。DataPoints

對於標記化查詢，每個維度清單中有三個項目：

- db.query_tokenized.statement— 標記化的查詢陳述式。
- db.query_tokenized.db_id — Performance Insights 為您產生的綜合 ID。此範例會傳回 pi-1064184600 合成 ID。

- `db.query_tokenized.id`– 績效詳情中查詢的 ID。

在 AWS Management Console 中，此 ID 稱為支援 ID。如此命名是因為 ID 是 AWS Support 可以檢查的資料，以協助您對資料庫的問題進行疑難排解。AWS 非常重視資料的安全性和隱私權，幾乎所有資料都會以您的 AWS KMS 客戶主金鑰 (CMK) 加密儲存。因此，AWS 中的任何人都無法查看這些資料。在先前的範例中，`tokenized.statement` 和 `tokenized.db_id` 都同時會以加密的形式存放。如果有資料庫相關的問題，AWS Support 可透過參考支援 ID 來提供協助。

進行查詢時，在 Group 中指定 GroupBy 可能會讓您省下不少心力。然而，如需對已傳回的資料進行更精細的控制，請指定維度的清單。例如，如果所需的是 `db.query_tokenized.statement`，則可將 Dimensions 屬性新增至 `query.json` 檔案。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": {
      "Group": "db.query_tokenized",
      "Dimensions": ["db.query_tokenized.statement"],
      "Limit": 10
    }
  }
]
```

擷取由查詢篩選的資料庫負載平均值

此範例中的對應 API 查詢與 [擷取最上層查詢的資料庫負載平均值](#) 中的命令類似。然而，查詢 `query.json` 檔案有以下內容。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.wait_state", "Limit": 5 },
    "Filter": { "db.query_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
  }
]
```

在此響應中，所有值都根據在 `query.json` 文件中指定的標記化查詢 `AKIAIOSFODNN7EXAMPLE` 的貢獻進行過濾。`AKIAIOSFODNN7EXAMPLE` 這些索引鍵也可能會遵循與沒有篩選器的查詢不同的順序，因為它是影響篩選查詢的前五個等待狀態。

亞馬遜 CloudWatch 指標的 Performance Insights

Performance Insights 會自動將指標發佈到 Amazon CloudWatch。您可以從「Performance Insights」中查詢相同的資料，但使用指標可 CloudWatch 讓您輕鬆新增 CloudWatch 警示。您也可以輕鬆將指標新增至現有的 CloudWatch 儀表板。

| 指標 | 描述 |
|-----------------|---|
| DBLoad | Amazon DocumentDB 的作用中工作階段數目。您通常會需要平均作用中工作階段數量的資料。在績效詳情中，系統會以 <code>db.load.avg</code> 的形式來查詢此資料。 |
| DBLoadCPU | 等待狀態類型為 CPU 的作用中階段作業數目。在 [Performance Insights] 中，會將此資料查詢為 <code>db.load.avg</code> ，並依等待狀態類型 CPU 篩選。 |
| 資料庫 LoadNon CPU | 等待狀態類型不是 CPU 的作用中階段作業數目。 |

Note

只有在資料庫執行個體有負載時，CloudWatch 才會將這些指標發佈到。

您可以使用 CloudWatch 主控台、或 CloudWatch API 來檢查這些指標。AWS CLI

例如，您可以執行 [get-metric-statistics](#) 命令來取得 DBLoad 測量結果的統計資料。

```
aws cloudwatch get-metric-statistics \  
  --region ap-south-1 \  
  --namespace AWS/DocDB \  
  --metric-name DBLoad \  
  --period 360 \  
  --statistics Average \  
  --start-time 2022-03-14T8:00:00Z \  
  --end-time 2022-03-14T9:00:00Z \  

```



```
--dimensions Name=DBInstanceIdentifier,Value=documentdbinstance
```

此範例會產生類似下列範例的輸出結果。

```
{
  "Datapoints": [
    {
      "Timestamp": "2022-03-14T08:42:00Z",
      "Average": 1.0,
      "Unit": "None"
    },
    {
      "Timestamp": "2022-03-14T08:24:00Z",
      "Average": 2.0,
      "Unit": "None"
    },
    {
      "Timestamp": "2022-03-14T08:54:00Z",
      "Average": 6.0,
      "Unit": "None"
    },
    {
      "Timestamp": "2022-03-14T08:36:00Z",
      "Average": 5.7,
      "Unit": "None"
    },
    {
      "Timestamp": "2022-03-14T08:06:00Z",
      "Average": 4.0,
      "Unit": "None"
    },
    {
      "Timestamp": "2022-03-14T08:00:00Z",
      "Average": 5.2,
      "Unit": "None"
    }
  ],
  "Label": "DBLoad"
}
```

您可以使用 CloudWatch 主控台內的 DB_PERF_INSIGHTS 指標數學函數來查詢 Amazon DocumentDB Performance Insights 計數器指標。此 DB_PERF_INSIGHTS 函數還包括以分鐘為間隔的 DBLoad 量度。

您可以在這些指標上設定 CloudWatch 警示。如需如何建立警示的詳細資訊，請參閱[從 AWS 資料庫建立 Performance Insights 計數器指標的警示](#)。

如需詳細資訊 CloudWatch，請參閱[什麼是 Amazon CloudWatch？](#) 在亞馬遜 CloudWatch 用戶指南。

計數器指標的 Performance Insights

計數器指標是「Performance Insights」儀表板中的作業系統度量。若要協助識別並分析效能問題，您可以將計數器指標與資料庫負載相互關聯。

績效詳情作業系統計數器

下列作業系統計數器可與 DocumentDB Performance Insights 一起使用。

| 計數器 | 類型 | 指標 |
|------------|----------------|--------------------------|
| 作用中 | memory | os.memory.active |
| buffers | memory | os.memory.buffers |
| cached | memory | os.memory.cached |
| dirty | memory | os.memory.dirty |
| free | memory | os.memory.free |
| inactive | memory | os.memory.inactive |
| mapped | memory | os.memory.mapped |
| pageTables | memory | os.memory.pageTables |
| slab | memory | os.memory.slab |
| total | memory | os.memory.total |
| writeback | memory | os.memory.writeback |
| idle | cpuUtilization | os.cpuUtilization.idle |
| system | cpuUtilization | os.cpuUtilization.system |

| 計數器 | 類型 | 指標 |
|----------|-------------------|----------------------------|
| total | cpuUtilization | os.cpuUtilization.total |
| user | cpuUtilization | os.cpuUtilization.user |
| wait | cpuUtilization | os.cpuUtilization.wait |
| one | loadAverageMinute | 操作系統。loadAverageMinute. 一 |
| fifteen | loadAverageMinute | 操作系統。loadAverageMinute. 十五 |
| five | loadAverageMinute | 操作系統。loadAverageMinute. 五 |
| cached | 交換 | os.swap.cached |
| free | 交換 | os.swap.free |
| in | 交換 | os.swap.in |
| out | 交換 | os.swap.out |
| total | 交換 | os.swap.total |
| rx | 網路 | os.network.rx |
| tx | 網路 | os.network.tx |
| numVCPUs | 一般 | os.general.numVCPUs |

使用 Amazon DocumentDB 開發

這些部分涵蓋了使用 Amazon DocumentDB (與 MongoDB 兼容性) 的開發。

主題

- [以編程方式連接 Amazon DocumentDB](#)
- [使用更改流與亞馬遜文檔數據庫](#)
- [使用AWS Lambda使用變更串流](#)
- [使用 JSON 結構定義驗證](#)
- [以複本集形式連接至 Amazon DocumentDB](#)
- [從亞馬遜 VPC 外部連接到 Amazon DocumentDB 集群](#)
- [從工作室 3T 連接到 Amazon DocumentDB 集群](#)
- [使用 Connect 到 Amazon DocumentDB DataGrip](#)
- [使用 Amazon EC2 Connect](#)
- [使用 Amazon DocumentDB JDBC 驅動程序進行 Connect](#)
- [使用 Amazon DocumentDB ODBC 驅動程序進行 Connect](#)

以編程方式連接 Amazon DocumentDB

本節包含的程式碼範例會示範如何使用多種不同的語言連線至 Amazon DocumentDB (具有 MongoDB 相容性)。範例會根據您連線的叢集已啟用或停用 Transport Layer Security (TLS)，分成兩個部分。根據預設，TLS 會在 Amazon DocumentDB 叢集上啟用。但如果您希望，您可以關閉 TLS。如需詳細資訊，請參閱 [對傳輸中資料進行加密](#)。

如果您嘗試從叢集所在 VPC 以外連接到 Amazon DocumentDB，請參閱[從亞馬遜 VPC 外部連接到 Amazon DocumentDB 集群](#)。

連線到您的叢集之前，您必須知道叢集是否已啟用 TLS。下節示範如何使用 AWS Management Console 或 AWS CLI判斷叢集的 `tls` 參數值。然後，您就可以尋找和套用適當的程式碼範例，繼續作業。

主題

- [判斷 tls 參數值](#)
- [使用已啟用的 TLS 連線](#)

- [使用已停用的 TLS 連線](#)

判斷 `tls` 參數值

判斷叢集是否已啟用 TLS 是兩個步驟的程序，您可以使用 AWS Management Console 或 AWS CLI 來執行。

1. 判斷管理您叢集的參數群組。

Using the AWS Management Console

1. [登入 AWS Management Console](https://console.aws.amazon.com/docdb)，然後在 <https://console.aws.amazon.com/docdb> 開啟 Amazon DocumentDB 主控台。
2. 在左側導覽窗格中選擇 Clusters (叢集)。
3. 在叢集清單中，選取您叢集的名稱。
4. 產生的頁面會顯示您所選取叢集的詳細資訊。向下捲動至 Cluster details (叢集詳細資訊)。在該區段底部的 Cluster parameter group (叢集參數群組) 下找到參數群組的名稱。

Using the AWS CLI

下列 AWS CLI 程式碼會決定管理叢集的參數。請確定以您叢集的名稱取代 `sample-cluster`。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

此操作的輸出如下所示：

```
[  
  [  
    "sample-cluster",  
    "sample-parameter-group"  
  ]  
]
```

2. 判斷您叢集參數群組的 `tls` 參數值。

Using the AWS Management Console

1. 在導覽窗格中，選擇 Parameter groups (參數群組)。
2. 在 Cluster parameter groups (叢集參數群組) 視窗中，選取您的叢集參數群組。
3. 產生的頁面會顯示您叢集參數群組的參數。您可以在這裡看到 `tls` 的參數值。如需修改此參數的資訊，請參閱[修改 Amazon DocumentDB 叢集參數群組](#)。

Using the AWS CLI

您可以使用命 `describe-db-cluster-parameters` AWS CLI 令來檢視叢集參數群組中參數的詳細資訊。

- **`--describe-db-cluster-parameters`**— 列出參數群組內的所有參數及其值。
- **`--db-cluster-parameter-group name`** – 必要。您叢集參數群組的名稱。

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的輸出如下所示：

```
{  
  "Parameters": [  
    {  
      "ParameterName": "profiler_threshold_ms",  
      "ParameterValue": "100",  
      "Description": "Operations longer than profiler_threshold_ms  
will be logged",  
      "Source": "system",  
      "ApplyType": "dynamic",  
      "DataType": "integer",  
      "AllowedValues": "50-2147483646",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot"  
    },  
    {  
      "ParameterName": "tls",  
      "ParameterValue": "disabled",  
      "Description": "Config to enable/disable TLS",
```

```
        "Source": "user",
        "ApplyType": "static",
        "DataType": "string",
        "AllowedValues": "disabled,enabled,fips-140-3",
        "IsModifiable": true,
        "ApplyMethod": "pending-reboot"
    }
]
}
```

Note

Amazon DocumentDB 支援 FIPS 140-3 端點，從 Amazon DocumentDB 5.0 (引擎版本 3.0.3727) 開始，這些區域中的叢集：ca-central-1、美國西部 2、us-east-1、us-east-2、-1、-1。us-gov-east us-gov-west

判斷 `tls` 參數值之後，請使用後續章節的其中一個程式碼範例來繼續連線您的叢集。

- [使用已啟用的 TLS 連線](#)
- [使用已停用的 TLS 連線](#)

使用已啟用的 TLS 連線

若要檢視程式碼範例，以程式設計方式連線至已啟用 TLS 的 Amazon DocumentDB 叢集，請為您要使用的語言選擇適當的索引標籤。

若要加密傳輸中的資料，請 `global-bundle.pem` 使用下列操作下載名為的 Amazon DocumentDB 公開金鑰。

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

若您的應用程式是位在 Microsoft Windows 上並且需要 PKCS7 檔案，您可以下載 PKCS7 憑證套件。此套件同時包含中繼憑證和根憑證 (下載連結：<https://truststore.pki.rds.amazonaws.com/global/global-bundle.p7b>)。

Python

下列程式碼示範如何在啟用 TLS 時使用 Python 連線到 Amazon DocumentDB。

```
import pymongo
import sys

##Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set
and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false')

##Specify the database to be used
db = client.sample_database

##Specify the collection to be used
col = db.sample_collection

##Insert a single document
col.insert_one({'hello':'Amazon DocumentDB'})

##Find the document that was previously written
x = col.find_one({'hello':'Amazon DocumentDB'})

##Print the result to the screen
print(x)

##Close the connection
client.close()
```

Node.js

下列程式碼示範如何在啟用 TLS 時使用 Node.js 連線到 Amazon DocumentDB。

```
var MongoClient = require('mongodb').MongoClient

//Create a MongoDB client, open a connection to DocDB; as a replica set,
// and specify the read preference as secondary preferred

var client = MongoClient.connect(
'mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
tls=true&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
{
  tlsCAFile: `global-bundle.pem` //Specify the DocDB; cert
},
```



```
function(err, client) {
  if(err)
    throw err;

  //Specify the database to be used
  db = client.db('sample-database');

  //Specify the collection to be used
  col = db.collection('sample-collection');

  //Insert a single document
  col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
    //Find the document that was previously written
    col.findOne({'hello':'DocDB;'}, function(err, result){
      //Print the result to the screen
      console.log(result);

      //Close the connection
      client.close()
    });
  });
});
```

PHP

下列程式碼示範如何在啟用 TLS 時使用 PHP 連線到 Amazon DocumentDB。

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

$TLS_DIR = "/home/ubuntu/global-bundle.pem";

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoClient("mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?retryWrites=false", ["tls" =>
  "true", "tlsCAFile" => $TLS_DIR ]);

//Specify the database and collection to be used
$col = $client->samledatabase->samplecollection;

//Insert a single document
$result = $col->insertOne( [ 'hello' => 'Amazon DocumentDB' ] );
```

```
//Find the document that was previously written
$result = $col->findOne(array('hello' => 'Amazon DocumentDB'));

//Print the result to the screen
print_r($result);
?>
```

Go

下列程式碼示範如何在啟用 TLS 時使用 Go 連線到 Amazon DocumentDB。

Note

從 1.2.1 版開始，MongoDB Go 驅動程式將只使用 `sslcertificateauthorityfile` 中找到的第一個 CA 伺服器憑證。下列範例程式碼透過手動將 `sslcertificateauthorityfile` 中找到的所有伺服器憑證附加到用戶端建立期間使用的自訂 TLS 組態中，來因應此限制。

```
package main

import (
    "context"
    "fmt"
    "log"
    "time"

    "go.mongodb.org/mongo-driver/bson"
    "go.mongodb.org/mongo-driver/mongo"
    "go.mongodb.org/mongo-driver/mongo/options"

    "io/ioutil"
    "crypto/tls"
    "crypto/x509"
    "errors"
)

const (
    // Path to the AWS CA file
    caFilePath = "global-bundle.pem"
```

```
// Timeout operations after N seconds
connectTimeout = 5
queryTimeout   = 30
username       = "<sample-user>"
password       = "<password>"
clusterEndpoint = "sample-cluster.node.us-east-1.docdb.amazonaws.com:27017"

// Which instances to read from
readPreference = "secondaryPreferred"

connectionStringTemplate = "mongodb://%s:%s@%s/sample-database?
tls=true&replicaSet=rs0&readpreference=%s"
)

func main() {

connectionURI := fmt.Sprintf(connectionStringTemplate, username, password,
clusterEndpoint, readPreference)

tlsConfig, err := getCustomTLSConfig(caFilePath)
if err != nil {
log.Fatalf("Failed getting TLS configuration: %v", err)
}

client, err :=
mongo.NewClient(options.Client().ApplyURI(connectionURI).SetTLSConfig(tlsConfig))
if err != nil {
log.Fatalf("Failed to create client: %v", err)
}

ctx, cancel := context.WithTimeout(context.Background(),
connectTimeout*time.Second)
defer cancel()

err = client.Connect(ctx)
if err != nil {
log.Fatalf("Failed to connect to cluster: %v", err)
}

// Force a connection to verify our connection string
err = client.Ping(ctx, nil)
if err != nil {
log.Fatalf("Failed to ping cluster: %v", err)
}
}
```

```
fmt.Println("Connected to DocumentDB!")

collection := client.Database("sample-database").Collection("sample-collection")

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

res, err := collection.InsertOne(ctx, bson.M{"name": "pi", "value": 3.14159})
if err != nil {
    log.Fatalf("Failed to insert document: %v", err)
}

id := res.InsertedID
log.Printf("Inserted document ID: %s", id)

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

cur, err := collection.Find(ctx, bson.D{})

if err != nil {
    log.Fatalf("Failed to run find query: %v", err)
}
defer cur.Close(ctx)

for cur.Next(ctx) {
    var result bson.M
    err := cur.Decode(&result)
    log.Printf("Returned: %v", result)

    if err != nil {
        log.Fatal(err)
    }
}

if err := cur.Err(); err != nil {
    log.Fatal(err)
}

}

func getCustomTLSConfig(caFile string) (*tls.Config, error) {
    tlsConfig := new(tls.Config)
```

```

certs, err := ioutil.ReadFile(caFile)

if err != nil {
    return tlsConfig, err
}

tlsConfig.RootCAs = x509.NewCertPool()
ok := tlsConfig.RootCAs.AppendCertsFromPEM(certs)

if !ok {
    return tlsConfig, errors.New("Failed parsing pem file")
}

return tlsConfig, nil

```

Java

從 Java 應用程式連線到啟用 TLS 的 Amazon DocumentDB 叢集時，您的程式必須使用 AWS 提供的憑證授權單位 (CA) 檔案來驗證連線。若要使用 Amazon RDS CA 憑證，請執行下列動作：

1. 從下載 Amazon RDS CA 文件 <https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem>。
2. 執行以下命令，建立將憑證授權機構憑證包含在檔案的信任存放區。請務必變更 `<truststorePassword>` 的內容。如果您正在存取同時包含舊憑證授權機構憑證 (rds-ca-2015-root.pem) 和新憑證授權機構憑證 (rds-ca-2019-root.pem) 的信任存放區，您可以將憑證套件匯入信任存放區。

以下範例 Shell 指令碼將憑證套件匯入 Linux 作業系統上的信任存放區。在下列範例中，將每個 `#####` 取代為您自己的資訊。最值得注意的是，無論示例目錄「`mydir`」位於腳本中的哪個位置，請將其替換為您為此任務創建的目錄。

```

mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
  ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/
  {split_after=1}{print > "rds-ca-" n ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do

```

```

    alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:;/; s/.*(CN=|CN = )//; print')
    echo "Importing $alias"
    keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
    rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
Alias | cut -d " " -f3- | while read alias
do
    expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
-alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*)\n/) { print
"$1\n"; }`
    echo " Certificate ${alias} expires in '$expiry'"
done

```

以下是範例 Shell 指令碼，會將憑證套件匯入 macOS 上的信任存放區。

```

mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
${mydir}/global-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-

for CERT in rds-ca-*; do
    alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:;/; s/.*(CN=|CN = )//; print')
    echo "Importing $alias"
    keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
    rm $CERT
done

rm ${mydir}/global-bundle.pem

```

```

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
Alias | cut -d " " -f3- | while read alias
do
    expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
    -alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*)\n/) { print
"$1\n"; }`
    echo " Certificate ${alias} expires in '$expiry'"
done

```

3. keystore在連線到 Amazon DocumentDB 叢集之前，請在應用程式中設定下列系統屬性，以便在程式中使用。

```

javax.net.ssl.trustStore: <truststore>
javax.net.ssl.trustStorePassword: <truststorePassword>

```

4. 下列程式碼示範如何在啟用 TLS 時使用 Java 連線到 Amazon DocumentDB。

```

package com.example.documentdb;

import com.mongodb.client.*;
import org.bson.Document;

public final class Test {
    private Test() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
ssl=true&replicaSet=rs0&readPreference=%s";
        String username = "<sample-user>";
        String password = "<password>";
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
        String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
clusterEndpoint, readPreference);

        String truststore = "<truststore>";
        String truststorePassword = "<truststorePassword>";

        System.setProperty("javax.net.ssl.trustStore", truststore);

```

```
        System.setProperty("javax.net.ssl.trustStorePassword",
truststorePassword);

        MongoClient mongoClient = MongoClient.create(connectionString);

        MongoDBDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection =
testDB.getCollection("sample-collection");

        Document doc = new Document("name", "pi").append("value", 3.14159);
numbersCollection.insertOne(doc);

        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        } finally {
            cursor.close();
        }
    }
}
```

C# / .NET

下列程式碼示範如何在啟用 TLS 時使用 C# / .NET 連線到 Amazon DocumentDB。

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using MongoDB.Driver;
using MongoDB.Bson;

namespace DocDB
{
    class Program
    {
        static void Main(string[] args)
```



```
{
    string template = "mongodb://{0}:{1}@{2}/sampledatabase?
tls=true&replicaSet=rs0&readpreference={3}";
    string username = "<sample-user>";
    string password = "<password>";
    string readPreference = "secondaryPreferred";
    string clusterEndpoint="sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
    string connectionString = String.Format(template, username, password,
clusterEndpoint, readPreference);

    string pathToCAFile = "<PATH/global-bundle.pem_file>";

    // ADD CA certificate to local trust store
    // DO this once - Maybe when your service starts
    X509Store localTrustStore = new X509Store(StoreName.Root);
    X509Certificate2Collection certificateCollection = new
X509Certificate2Collection();
    certificateCollection.Import(pathToCAFile);
    try
    {
        localTrustStore.Open(OpenFlags.ReadWrite);
        localTrustStore.AddRange(certificateCollection);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Root certificate import failed: " + ex.Message);
        throw;
    }
    finally
    {
        localTrustStore.Close();
    }

    var settings = MongoClientSettings.FromUrl(new
MongoUrl(connectionString));
    var client = new MongoClient(settings);

    var database = client.GetDatabase("sampledatabase");
    var collection =
database.GetCollection<BsonDocument>("samplecollection");
    var docToInsert = new BsonDocument { { "pi", 3.14159 } };
    collection.InsertOne(docToInsert);
}
```

```
}  
}
```

mongo shell

下列程式碼示範如何在啟用 TLS 時使用蒙戈殼層連接和查詢 Amazon DocumentDB。

1. Connect 到 Amazon DocumentDB 與蒙戈外殼。如果您使用的是早於 4.2 的 mongo shell 版本，請使用以下代碼進行連接。

```
mongo --ssl --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 --  
sslCAFile global-bundle.pem --username <sample-user> --password <password>
```

如果您使用的版本等於或大於 4.2，請使用以下代碼進行連接。AWS DocumentDB 不支援可重試寫入。例外：如果您使用的是 mongo shell，請不要在任何代碼字符串中包含該 `retryWrites=false` 命令。根據預設，會停用可重試寫入。包括 `retryWrites=false` 可能會導致正常讀取命令失敗。

```
mongo --tls --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 --  
tlsCAFile global-bundle.pem --username <sample-user> --password <password>
```

2. 插入單一文件。

```
db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
```

3. 尋找之前插入的文件。

```
db.myTestCollection.find({'hello':'Amazon DocumentDB'})
```

R

下面的代碼演示了如何使用蒙金石 (<https://jeroen.github.io/mongolite/>) 連接到 Amazon DocumentDB 與 R 時啟用 TLS。

```
#Include the mongolite library.  
library(mongolite)  
  
mongourl <- paste("mongodb://<sample-user>:<password>@sample-cluster.node.us-  
east-1.docdb.amazonaws.com:27017/test2?ssl=true&",  
                "readPreference=secondaryPreferred&replicaSet=rs0", sep="")
```

```
#Create a MongoDB client, open a connection to Amazon DocumentDB as a replica
# set and specify the read preference as secondary preferred
client <- mongo(url = mongourl, options = ssl_options(weak_cert_validation = F, ca
  = "<PATH/global-bundle.pem>"))

#Insert a single document
str <- c('{"hello" : "Amazon DocumentDB"}')
client$insert(str)

#Find the document that was previously written
client$find()
```

Ruby

下列程式碼示範如何在啟用 TLS 時使用紅寶石連線到 Amazon DocumentDB 料庫。

```
require 'mongo'
require 'neatjson'
require 'json'
client_host = 'mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017'
client_options = {
  database: 'test',
  replica_set: 'rs0',
  read: {:secondary_preferred => 1},
  user: '<sample-user>',
  password: '<password>',
  ssl: true,
  ssl_verify: true,
  ssl_ca_cert: '<PATH/global-bundle.pem>',
  retry_writes: false
}

begin
  ##Create a MongoDB client, open a connection to Amazon DocumentDB as a
  ## replica set and specify the read preference as secondary preferred
  client = Mongo::Client.new(client_host, client_options)

  ##Insert a single document
  x = client[:test].insert_one({"hello":"Amazon DocumentDB"})

  ##Find the document that was previously written
  result = client[:test].find()
```

```
#Print the document
result.each do |document|
  puts JSON.neat_generate(document)
end
end

#Close the connection
client.close
```

使用已停用的 TLS 連線

若要檢視程式碼範例，以程式設計方式連線至已停用 TLS 的 Amazon DocumentDB 叢集，請選擇您要使用的語言索引標籤。

Python

下面的代碼演示如何連接到 Amazon DocumentDB 使用 Python 時 TLS 被禁用。

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set
and specify the read preference as secondary preferred

import pymongo
import sys

client = pymongo.MongoClient('mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false')

##Specify the database to be used
db = client.sample_database

##Specify the collection to be used
col = db.sample_collection

##Insert a single document
col.insert_one({'hello':'Amazon DocumentDB'})

##Find the document that was previously written
x = col.find_one({'hello':'Amazon DocumentDB'})

##Print the result to the screen
```

```
print(x)

##Close the connection
client.close()
```

Node.js

下面的代碼演示如何連接到 Amazon DocumentDB 使用 Node.js 時 TLS 被禁用。

```
var MongoClient = require('mongodb').MongoClient;

//Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set,
// and specify the read preference as secondary preferred
var client = MongoClient.connect(
  'mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
  {
    useNewUrlParser: true
  },

function(err, client) {
  if(err)
    throw err;
  //Specify the database to be used
  db = client.db('sample-database');

  //Specify the collection to be used
  col = db.collection('sample-collection');

  //Insert a single document
  col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
    //Find the document that was previously written
    col.findOne({'hello':'Amazon DocumentDB'}, function(err, result){
      //Print the result to the screen
      console.log(result);

      //Close the connection
      client.close()
    });
  });
});
```

PHP

下面的代碼演示如何連接到 Amazon DocumentDB 使用 PHP 時 TLS 被禁用。

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoClient("mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?retryWrites=false");

//Specify the database and collection to be used
$col = $client->sampldatabase->samplecollection;

//Insert a single document
$result = $col->insertOne( [ 'hello' => 'Amazon DocumentDB' ] );

//Find the document that was previously written
$result = $col->findOne(array('hello' => 'Amazon DocumentDB'));

//Print the result to the screen
print_r($result);
?>
```

Go

下列程式碼示範如何在停用 TLS 時使用 Go 連線到 Amazon DocumentDB。

```
package main

import (
    "context"
    "fmt"
    "log"
    "time"

    "go.mongodb.org/mongo-driver/bson"
    "go.mongodb.org/mongo-driver/mongo"
    "go.mongodb.org/mongo-driver/mongo/options"
)

const (
    // Timeout operations after N seconds
```

```
connectTimeout = 5
queryTimeout   = 30
username       = "<sample-user>"
password       = "<password>"
clusterEndpoint = "sample-cluster.node.us-east-1.docdb.amazonaws.com:27017"

// Which instances to read from
readPreference = "secondaryPreferred"
connectionStringTemplate = "mongodb://%s:%s@%s/sample-database?
replicaSet=rs0&readpreference=%s"
)

func main() {

    connectionURI := fmt.Sprintf(connectionStringTemplate, username, password,
    clusterEndpoint, readPreference)

    client, err := mongo.NewClient(options.Client().ApplyURI(connectionURI))
    if err != nil {
        log.Fatalf("Failed to create client: %v", err)
    }

    ctx, cancel := context.WithTimeout(context.Background(),
    connectTimeout*time.Second)
    defer cancel()

    err = client.Connect(ctx)
    if err != nil {
        log.Fatalf("Failed to connect to cluster: %v", err)
    }

    // Force a connection to verify our connection string
    err = client.Ping(ctx, nil)
    if err != nil {
        log.Fatalf("Failed to ping cluster: %v", err)
    }

    fmt.Println("Connected to DocumentDB!")

    collection := client.Database("sample-database").Collection("sample-collection")

    ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
    defer cancel()
}
```

```
res, err := collection.InsertOne(ctx, bson.M{"name": "pi", "value": 3.14159})
if err != nil {
    log.Fatalf("Failed to insert document: %v", err)
}

id := res.InsertedID
log.Printf("Inserted document ID: %s", id)

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

cur, err := collection.Find(ctx, bson.D{})

if err != nil {
    log.Fatalf("Failed to run find query: %v", err)
}
defer cur.Close(ctx)

for cur.Next(ctx) {
    var result bson.M
    err := cur.Decode(&result)
    log.Printf("Returned: %v", result)

    if err != nil {
        log.Fatal(err)
    }
}

if err := cur.Err(); err != nil {
    log.Fatal(err)
}
}
```

Java

下面的代碼演示如何連接到 Amazon DocumentDB 使用 Java 時 TLS 被禁用。

```
package com.example.documentdb;

import com.mongodb.MongoClient;
import com.mongodb.MongoClientURI;
import com.mongodb.ServerAddress;
import com.mongodb.MongoException;
```



```
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.MongoCollection;
import org.bson.Document;

public final class Main {
    private Main() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
replicaSet=rs0&readpreference=%s";
        String username = "<sample-user>";
        String password = "<password>";
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
        String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
clusterEndpoint, readPreference);

        MongoClientURI clientURI = new MongoClientURI(connectionString);
        MongoClient mongoClient = new MongoClient(clientURI);

        MongoDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection = testDB.getCollection("sample-
collection");

        Document doc = new Document("name", "pi").append("value", 3.14159);
        numbersCollection.insertOne(doc);

        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        } finally {
            cursor.close();
        }
    }
}
```

C# / .NET

下列程式碼示範如何在停用 TLS 時使用 C# / .NET 連線到 Amazon DocumentDB。

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using MongoDB.Driver;
using MongoDB.Bson;

namespace CSharpSample
{
    class Program
    {
        static void Main(string[] args)
        {
            string template = "mongodb://{0}:{1}@{2}/sampledatabase?
replicaSet=rs0&readpreference={3}";
            string username = "<sample-user>";
            string password = "<password>";
            string clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
            string readPreference = "secondaryPreferred";
            string connectionString = String.Format(template, username, password,
clusterEndpoint, readPreference);

            var settings = MongoClientSettings.FromUrl(new
MongoUrl(connectionString));
            var client = new MongoClient(settings);

            var database = client.GetDatabase("sampledatabase");
            var collection =
database.GetCollection<BsonDocument>("samplecollection");
            var docToInsert = new BsonDocument { { "pi", 3.14159 } };
            collection.InsertOne(docToInsert);
        }
    }
}
```

mongo shell

下列程式碼示範如何在停用 TLS 時使用蒙戈殼層連接和查詢 Amazon DocumentDB。

1. Connect 到 Amazon DocumentDB 與蒙戈外殼。

```
mongo --host mycluster.node.us-east-1.docdb.amazonaws.com:27017 --  
username <sample-user> --password <password>
```

2. 插入單一文件。

```
db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
```

3. 尋找之前插入的文件。

```
db.myTestCollection.find({'hello':'Amazon DocumentDB'})
```

R

下面的代碼演示了如何連接到 Amazon DocumentDB 與 R 使用蒙金石 (<https://jeroen.github.io/mongolite/>) 當 TLS 被禁用。

```
#Include the mongolite library.  
library(mongolite)  
  
#Create a MongoDB client, open a connection to Amazon DocumentDB as a replica  
# set and specify the read preference as secondary preferred  
client <- mongo(url = "mongodb://<sample-user>:<password>@sample-  
cluster.node.us-east-1.docdb.amazonaws.com:27017/sample-database?  
readPreference=secondaryPreferred&replicaSet=rs0")  
  
##Insert a single document  
str <- c('{"hello" : "Amazon DocumentDB"}')  
client$insert(str)  
  
##Find the document that was previously written  
client$find()
```

Ruby

下列程式碼示範如何在停用 TLS 時使用紅寶石連線到 Amazon DocumentDB 料庫。

```
require 'mongo'
require 'neatjson'
require 'json'
client_host = 'mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017'
client_options = {
  database: 'test',
  replica_set: 'rs0',
  read: {:secondary_preferred => 1},
  user: '<sample-user>',
  password: '<password>',
  retry_writes: false
}

begin
  ##Create a MongoDB client, open a connection to Amazon DocumentDB as a
  ## replica set and specify the read preference as secondary preferred
  client = Mongo::Client.new(client_host, client_options)

  ##Insert a single document
  x = client[:test].insert_one({"hello":"Amazon DocumentDB"})

  ##Find the document that was previously written
  result = client[:test].find()

  #Print the document
  result.each do |document|
    puts JSON.neat_generate(document)
  end
end

#Close the connection
client.close
```

使用更改流與亞馬遜文檔數據庫

Amazon DocumentDB 中的變更串流功能 (與 MongoDB 相容性) 可提供叢集集中發生的變更事件的時間順序序列。您可以從變更串流讀取事件，以實作各種不同的使用案例，包括下列各項：

- 變更通知
- 使用亞馬遜進行全文搜索 OpenSearch 服務項目 (OpenSearch 服務)

- [使用亞馬遜紅移進行分析](#)

應用程式可以使用變更串流來訂閱個別集合上的資料變更。變更串流事件發生時便會排序這些事件，並在記錄事件後儲存這些事件 3 小時 (預設)。保留期最多可以延長 7 天 `change_stream_log_retention_duration` 參數。如要修改串流保留期限的變更，請參閱 [修改變更串流記錄保留期間](#)。

主題

- [受支援的 操作](#)
- [帳單](#)
- [限制](#)
- [啟用變更串流](#)
- [範例：搭配 Python 使用變更串流](#)
- [完整文件查閱](#)
- [繼續變更串流](#)
- [使用恢復變更串流startAtOperationTime](#)
- [變更串流中的交易](#)
- [修改變更串流日誌保留期間](#)

受支援的 操作

亞馬遜文檔數據庫支持變更流的以下操作：

- MongoDB 中支援的所有變更事件 `db.collection.watch()`、`db.watch()` 和 `client.watch()` API。
- 更新的完整文件查閱。
- 彙總階段：`$match`、`$project`、`$redact`，以及 `$addField` 和 `$replaceRoot`。
- 從恢復令牌恢復更改流
- 使用從時間戳記恢復更改流 `startAtOperation` (適用於亞馬遜文檔 DB v4.0 +)

帳單

Amazon DocumentDB 變更串流功能預設為停用狀態，且在啟用此功能之前不會產生任何額外費用。在叢集中使用變更串流會產生額外的讀取和寫入 I/O 和儲存費用。您可以使

用 `modifyChangeStreamsAPI` 操作可為您的叢集啟用此功能。如需定價的詳細資訊，請參閱 [亞馬遜文檔數據庫定價](#)。

限制

變更串流在 Amazon 文件資料庫中有下列限制：

- 變更串流只能從連線到 Amazon DocumentDB 叢集的主要執行個體開啟。目前不支援從複本執行個體上的變更串流讀取。呼叫 `watch()` API 操作時，您必須指定 **primary** 讀取偏好設定，以確保所有讀取全都導向至主要執行個體 (請參閱 [範例](#) 一節)。
- 寫入集合變更串流的事件最多可使用 7 天 (預設值為 3 小時)。在日誌保留期間時段後將會刪除變更串流資料，即便未發生任何新變更。
- 在如 `updateMany` 或 `deleteMany` 等集合上長時間執行寫入操作時，會暫停寫入變更串流事件，直到長時間執行的寫入操作完成為止。
- 亞馬遜文檔數據庫不支持 MongoDB 操作日誌 (`oplog`)。
- 使用亞馬遜 DocumentDB 時，您必須明確啟用指定集合上的變更串流。
- 如果變更串流事件的總大小 (如有要求，包括變更資料與完整文件) 大於 16 MB，用戶端將會遇到變更串流讀取失敗的情況。
- 使用時目前不支援 Ruby 驅動程式 `db.watch()` 和 `client.watch()` 與亞馬遜文檔數據庫 v3.6。

啟用變更串流

您可以為指定資料庫內的所有集合啟用 Amazon DocumentDB 變更串流，或僅針對選取的集合啟用。以下是針對使用 mongo 殼層的不同使用案例，如何啟用變更串流的範例。指定資料庫和集合名稱時，空的字串會被視為萬用字元。

```
//Enable change streams for the collection "foo" in database "bar"
db.adminCommand({modifyChangeStreams: 1,
  database: "bar",
  collection: "foo",
  enable: true});
```

```
//Disable change streams on collection "foo" in database "bar"
db.adminCommand({modifyChangeStreams: 1,
  database: "bar",
  collection: "foo",
  enable: false});
```

```
//Enable change streams for all collections in database "bar"  
db.adminCommand({modifyChangeStreams: 1,  
  database: "bar",  
  collection: "",  
  enable: true});
```

```
//Enable change streams for all collections in all databases in a cluster  
db.adminCommand({modifyChangeStreams: 1,  
  database: "",  
  collection: "",  
  enable: true});
```

如果下列任一項成立，集合將會啟用變更串流：

- 明確地啟用資料庫和集合。
- 已啟用包含集合的資料庫。
- 已啟用所有資料庫。

如果父資料庫也啟用變更串流或叢集內的所有資料庫都已啟用，捨棄資料庫的集合並不會停用該集合的變更串流。如果新的集合以遭刪除集合的相同名稱建立，該集合將會啟用變更串流。

您可以使用 `$listChangeStreams` 彙總管線階段列出叢集所有已啟用的變更串流。Amazon DocumentDB 支援的所有彙總階段都可用於管道中進行其他處理。如果已停用之前啟用的集合，該集合將不會在 `$listChangeStreams` 輸出中出現。

```
//List all databases and collections with change streams enabled  
cursor = new DBCommandCursor(db,  
  db.runCommand(  
    {aggregate: 1,  
    pipeline: [{ $listChangeStreams: 1 }],  
    cursor: { } }));
```

```
//List of all databases and collections with change streams enabled  
{ "database" : "test", "collection" : "foo" }  
{ "database" : "bar", "collection" : "" }  
{ "database" : "", "collection" : "" }
```

```
//Determine if the database "bar" or collection "bar.foo" have change streams enabled  
cursor = new DBCommandCursor(db,
```

```

db.runCommand(
  {aggregate: 1,
    pipeline: [{$listChangeStreams: 1},
      {$match: {$or: [{database: "bar", collection: "foo"},
        {database: "bar", collection: ""},
        {database: "", collection: ""}]}}
    ],
  cursor: {}}));

```

範例：搭配 Python 使用變更串流

以下是在集合級別使用亞馬遜文檔數據庫更改流與 Python 的示例。

```

import os
import sys
from pymongo import MongoClient, ReadPreference

username = "DocumentDBusername"
password = <Insert your password>

clusterendpoint = "DocumentDBClusterEndpoint"
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
  tlsCAFile='global-bundle.pem')

db = client['bar']

#While 'Primary' is the default read preference, here we give an example of
#how to specify the required read preference when reading the change streams
coll = db.get_collection('foo', read_preference=ReadPreference.PRIMARY)
#Create a stream object
stream = coll.watch()
#Write a new document to the collection to generate a change event
coll.insert_one({'x': 1})
#Read the next change event from the stream (if any)
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},

```



```
'operationType': 'insert'}
''''

#A subsequent attempt to read the next change event returns nothing, as there are no
new changes
print(stream.try_next())

''''
Expected Output:
None
''''

#Generate a new change event by updating a document
result = coll.update_one({'x': 1}, {'$set': {'x': 2}})
print(stream.try_next())

''''
Expected Output:
{'_id': {'_data': '015daf99d400000001010000000100009025'},
'clusterTime': Timestamp(1571789268, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 2}}}
''''
```

以下是在資料庫層級搭配 Python 使用亞馬遜文件資料庫變更串流的範例。

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint"
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
    tlsCAFile='global-bundle.pem')

db = client['bar']
#Create a stream object
stream = db.watch()
coll = db.get_collection('foo')
#Write a new document to the collection foo to generate a change event
```

```
coll.insert_one({'x': 1})

#Read the next change event from the stream (if any)
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'insert'}
"""

#A subsequent attempt to read the next change event returns nothing, as there are no
new changes
print(stream.try_next())

"""
Expected Output:
None
"""

coll = db.get_collection('foo1')

#Write a new document to another collection to generate a change event
coll.insert_one({'x': 1})
print(stream.try_next())

"""
Expected Output: Since the change stream cursor was the database level you can see
change events from different collections in the same database
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo1', 'db': 'bar'},
'operationType': 'insert'}
"""
```

完整文件查閱

更新變更事件不包括完整文件；只包含所做的變更。如果您的使用案例需要受更新影響的完整文件，您可以在開啟串流時啟用完整文件查閱。

更新變更串流事件的 `fullDocument` 文件代表在文件查閱時已更新文件最新的版本。如果更新操作與 `fullDocument` 查閱之間發生變更，`fullDocument` 文件可能不會代表更新時的文件狀態。

```
#Create a stream object with update lookup enabled
stream = coll.watch(full_document='updateLookup')

#Generate a new change event by updating a document
result = coll.update_one({'x': 2}, {'$set': {'x': 3}})

stream.try_next()

#Output:
{'_id': {'_data': '015daf9b7c00000001010000000100009025'},
 'clusterTime': Timestamp(1571789692, 1),
 'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
 'fullDocument': {'_id': ObjectId('5daf9502ea258751778163d7'), 'x': 3},
 'ns': {'coll': 'foo', 'db': 'bar'},
 'operationType': 'update',
 'updateDescription': {'removedFields': [], 'updatedFields': {'x': 3}}}
```

繼續變更串流

您稍後可以使用繼續字符 (等於上次擷取之變更事件文件的 `_id` 欄位) 繼續變更串流。

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint"
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
  tlsCAFile='global-bundle.pem', retryWrites='false')

db = client['bar']
coll = db.get_collection('foo')
#Create a stream object
```

```

stream = db.watch()
coll.update_one({'x': 1}, {'$set': {'x': 4}})
event = stream.try_next()
token = event['_id']
print(token)

"""
Output: This is the resume token that we will later use to resume the change stream
{'_data': '015daf9c5b00000001010000000100009025'}
"""

#Python provides a nice shortcut for getting a stream's resume token
print(stream.resume_token)

"""
Output
{'_data': '015daf9c5b00000001010000000100009025'}
"""

#Generate a new change event by updating a document
result = coll.update_one({'x': 4}, {'$set': {'x': 5}})
#Generate another change event by inserting a document
result = coll.insert_one({'y': 5})
#Open a stream starting after the selected resume token
stream = db.watch(full_document='updateLookup', resume_after=token)
#Our first change event is the update with the specified _id
print(stream.try_next())

"""
#Output: Since we are resuming the change stream from the resume token, we will see all
events after the first update operation. In our case, the change stream will resume
from the update operation {x:5}

{'_id': {'_data': '015f7e8f0c000000060100000006000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602129676, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0ac423bafb9adba2')},
'fullDocument': {'_id': ObjectId('5f7e8f0ac423bafb9adba2'), 'x': 5},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""

#Followed by the insert
print(stream.try_next())

"""
#Output:

```

```
{'_id': {'_data': '015f7e8f0c000000070100000007000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602129676, 7),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94')},
'fullDocument': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94'), 'y': 5}}
"""
```

使用恢復變更串流startAtOperationTime

您可以稍後從特定的時間戳記恢復變更串流，方法是使用startAtOperationTime。

Note

使用的能力startAtOperationTime是在亞馬遜文檔 DB 4.0+ 可用。使用時startAtOperationTime，變更串流游標只會傳回在指定時間戳記或之後發生的變更。該startAtOperationTime和resumeAfter命令是相互排斥的，因此不能一起使用。

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint"
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
  tlsCAFile='rds-root-ca-2020.pem',retryWrites='false')
db = client['bar']
coll = db.get_collection('foo')
#Create a stream object
stream = db.watch()
coll.update_one({'x': 1}, {'$set': {'x': 4}})
event = stream.try_next()
timestamp = event['clusterTime']
print(timestamp)
"""

Output
Timestamp(1602129114, 4)
"""

#Generate a new change event by updating a document
result = coll.update_one({'x': 4}, {'$set': {'x': 5}})
```

```
result = coll.insert_one({'y': 5})
#Generate another change event by inserting a document
#Open a stream starting after specified time stamp

stream = db.watch(start_at_operation_time=timestamp)
print(stream.try_next())

"""
#Output: Since we are resuming the change stream at the time stamp of our first update
operation (x:4), the change stream cursor will point to that event
{'_id': {'_data': '015f7e941a000000030100000003000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602130970, 3),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e9417c423bafb9adbb1')},
'updateDescription': {'updatedFields': {'x': 4}, 'removedFields': []}}
"""

print(stream.try_next())
"""
#Output: The second event will be the subsequent update operation (x:5)
{'_id': {'_data': '015f7e9502000000050100000005000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602131202, 5),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e94ffc423bafb9adbb2')},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""

print(stream.try_next())

"""
#Output: And finally the last event will be the insert operation (y:5)
{'_id': {'_data': '015f7e9502000000060100000006000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602131202, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e95025c4a569e0f6dde92')},
'fullDocument': {'_id': ObjectId('5f7e95025c4a569e0f6dde92'), 'y': 5}}
"""
```

變更串流中的交易

變更串流事件不會包含來自未提交和/或中止交易的事件。例如，如果您使用一個交易開始INSERT操作和一個UPDATE操作。如果您的INSERT作業成功，但UPDATE操作失敗，事務將被回滾。由於此交易已復原，因此您的變更串流將不會包含此交易的任何事件。

修改變更串流日誌保留期間

您可以將串流記錄保留期間的變更修改為 1 小時到 7 天之間，AWS Management Console或AWS CLI。

Using the AWS Management Console

修改變更串流日誌保留期限

1. 登入AWS Management Console，然後在以下位置打開亞馬遜文檔數據庫控制台<https://console.aws.amazon.com/docdb>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

Tip


如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

3. 在 Parameter groups (參數群組) 窗格中，選擇與叢集相關聯的叢集參數群組。若要識別與叢集相關聯的叢集參數群組，請參閱[確定 Amazon DocumentDB 集群的參數組](#)。
4. 結果頁面會顯示這個叢集參數群組的參數，及其對應詳細資訊。選取 `change_stream_log_retention_duration` 參數。
5. 在頁面右上角，選擇 Edit (編輯) 以變更此參數值。
該`change_stream_log_retention_duration`參數可以修改為 1 小時到 7 天之間。
6. 進行變更，然後選擇 Modify cluster parameter (修改叢集參數) 以儲存變更。若要捨棄變更，請選擇 Cancel (取消)。

Using the AWS CLI

若要修改叢集參數群組的 `change_stream_log_retention_duration` 參數，請使用 `modify-db-cluster-parameter-group` 操作搭配下列參數：

- **--db-cluster-parameter-group-name** – 必要。您想要修改的叢集參數群組名稱。若要識別與叢集相關聯的叢集參數群組，請參閱[確定 Amazon DocumentDB 集群的參數組](#)。
- **--parameters** – 必要。您要修改的參數。每個參數項目必須包括下列：
 - **ParameterName**— 您要修改的參數名稱。在這種情況下，它是 `change_stream_log_retention_duration`
 - **ParameterValue**— 此參數的新值。
 - **ApplyMethod**— 套用此參數變更的方式。允許值為 `immediate` 和 `pending-reboot`。

 Note

`static` 必須擁有使用 `ApplyType` 的參數 `ApplyMethod pending-reboot`。

1. 若要變更參數 `change_stream_log_retention_duration` 的值，請執行下列命令，並以您想要修改參數的值取代 `parameter-value`。

若為 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group \  
  --parameters  
  "ParameterName=change_stream_log_retention_duration,ParameterValue=<parameter-  
value>,ApplyMethod=immediate"
```

針對 Windows：

```
aws docdb modify-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name sample-parameter-group ^  
  --parameters  
  "ParameterName=change_stream_log_retention_duration,ParameterValue=<parameter-  
value>,ApplyMethod=immediate"
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "DBClusterParameterGroupName": "sample-parameter-group"  
}
```


2. 等待至少 5 分鐘。
3. 列出 `sample-parameter-group` 的參數值確保已進行變更。

若為 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name sample-parameter-group
```

針對 Windows：

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{  
  "Parameters": [  
    {  
      "ParameterName": "audit_logs",  
      "ParameterValue": "disabled",  
      "Description": "Enables auditing on cluster.",  
      "Source": "system",  
      "ApplyType": "dynamic",  
      "DataType": "string",  
      "AllowedValues": "enabled,disabled",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot"  
    },  
    {  
      "ParameterName": "change_stream_log_retention_duration",  
      "ParameterValue": "12345",  
      "Description": "Duration of time in seconds that the change stream  
log is retained and can be consumed.",  
      "Source": "user",  
      "ApplyType": "dynamic",  
      "DataType": "integer",  
      "AllowedValues": "3600-86400",  
      "IsModifiable": true,  
      "ApplyMethod": "immediate"  
    }  
  ]  
}
```

```
}
```

Note

變更串流記錄保留不會刪除超過設定的記錄

檔 `change_stream_log_retention_duration` 直到記錄檔大小大於 (>) 51,200MB 為止的值。

使用AWS Lambda使用變更串流

亞馬遜文檔數據庫集成AWS Lambda這樣您就可以使用 Lambda 函數來處理變更串流中的記錄。Lambda 事件來源對應是一種資源，可用來叫用 Lambda 函數，以處理不直接叫用 Lambda 的亞馬遜文件資料庫事件。使用 Amazon DocumentDB 變更串流做為事件來源，您可以建立事件驅動的應用程式，以回應資料中的變更。例如，您可以使用 Lambda 函數來處理新文件、追蹤現有文件的更新，或記錄已刪除的文件。

您可以設定事件來源對應，將記錄從 Amazon 文件資料庫變更串流傳送至 Lambda 函數。事件可以一次傳送一個或批次傳送以提高效率，並按順序處理。您可以根據特定的時間範圍持續時間 (0-300 秒) 或批次記錄計數 (最多 10,000 筆記錄) 來設定事件來源對應的批次處理行為。您可以建立多個事件來源對應，以處理具有多個 Lambda 函數的相同資料，或使用單一函數處理來自多個串流的不同項目。

如果您的函數傳回錯誤，Lambda 會重試批次，直到成功處理為止。如果變更串流中的事件已過期，Lambda 將停用事件來源對應。在這種情況下，您可以建立新的事件來源對應，並使用您選擇的起始位置進行配置。由於輪詢器的分散式特性，Lambda 事件來源映射會至少處理一次事件。因此在極少數情況下，您的 Lambda 函數可能會收到重複的事件。遵循使用的最佳做法AWS Lambda函數和構建冪等函數，以避免與重複事件有關的問題。如需詳細資訊，請參[使用AWS Lambda console與亞馬遜文檔數據庫](#)在AWS Lambda開發者指南。

作為效能最佳實務，Lambda 函數必須為短期函數。為了避免產生不必要的處理延遲，此函數也不應執行複雜的邏輯。尤其對於高速串流而言，相較於同步長時間執行的 Lambda，觸發非同步後續處理 Step Function 工作流程是較佳的做法。如需 AWS Lambda 的詳細資訊，請參閱《[AWS Lambda 開發人員指南](#)》。

限制

以下是使用亞馬遜文檔數據庫和時要考慮的限制AWS Lambda:

- AWS Lambda目前僅在亞馬遜文檔數據庫 4.0 和 5.0 上受支持。
- AWS Lambda彈性叢集或全域叢集目前不支援。
- AWS Lambda有效載荷大小不能超過 6MB。如需 Lambda 批次大小的詳細資訊，請參閱中的「批次處理行為」[事件來源映射](#)區段中的AWS Lambda開發者指南。

使用 JSON 結構定義驗證

使用\$jsonSchema評估查詢運算子，您可以驗證要插入集合的文件。

主題

- [建立和使用 JSON 結構描述驗證](#)
- [支援關鍵字](#)
- [限制](#)

建立和使用 JSON 結構描述驗證

使用結構描述驗證建立集合

您可以使用createCollection作業和驗證規則建立集合。這些驗證規則會在插入或更新 Amazon DocumentDB 文件期間套用。下列程式碼範例顯示員工集合的驗證規則：

```
db.createCollection("employees", {
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "employee validation",
      "required": [ "name", "employeeId"],
      "properties": {
        "name": {
          "bsonType": "object",
          "properties": {
            "firstName": {
              "bsonType": ["string"]
            },
            "lastName": {
              "bsonType": ["string"]
            }
          }
        }
      }
    }
  },
},
```

```
        "additionalProperties" : false
    },
    "employeeId": {
        "bsonType": "string",
        "description": "Unique Identifier for employee"
    },
    "salary": {
        "bsonType": "double"
    },
    "age": {
        "bsonType": "number"
    }
},
"additionalProperties" : true
}
},
"validationLevel": "strict", "validationAction": "error"
} )
```

插入有效的文件

下列範例會插入符合上述結構描述驗證規則的文件：

```
db.employees.insert({"name" : { "firstName" : "Carol" , "lastName" : "Smith"},
"employeeId": "c720a" , "salary": 1000.0 })
db.employees.insert({ "name" : { "firstName" : "William", "lastName" : "Taylor" },
"employeeId" : "c721a", "age" : 24})
```

插入無效的文件

下列範例會插入不符合上述結構描述驗證規則的文件。在這個例子中，員工 ID 值不是一個字符串：

```
db.employees.insert({
    "name" : { "firstName" : "Carol" , "lastName" : "Smith"},
    "employeeId": 720 ,
    "salary": 1000.0
})
```

此範例顯示文件中的語法不正確。

修改集合

該collMod命令用於添加或修改現有集合的驗證規則。下面的例子添加了一個工資字段的必填字段列表：

```
db.runCommand({"collMod" : "employees",
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "employee validation",
      "required": [ "name", "employeeId", "salary"],
      "properties": {
        "name": {
          "bsonType": "object",
          "properties": {
            "firstName": {
              "bsonType": ["string"]
            },
            "lastName": {
              "bsonType": ["string"]
            }
          },
          "additionalProperties" : false
        },
        "employeeId": {
          "bsonType": "string",
          "description": "Unique Identifier for employee"
        },
        "salary": {
          "bsonType": "double"
        },
        "age": {
          "bsonType": "number"
        }
      },
      "additionalProperties" : true
    }
  }
})
```

處理驗證規則變更前新增的文件

若要處理在變更驗證規則之前新增至您收集的文件，請使用下列validationLevel修正因子：

- **strict** : 對所有插入和更新套用驗證規則。
- **中度** : 將驗證規則套用至現有有效文件。在更新期間，不會檢查現有的無效文件。

在下列範例中，更新名為「employee」集合的驗證規則之後，薪資欄位是必要的。更新下列文件將會失敗：

```
db.runCommand({
  update: "employees",
  updates: [{
    q: { "employeeId": "c721a" },
    u: { age: 25 , salary : 1000},
    upsert: true }]
})
```

Amazon DocumentDB 返回以下輸出：

```
{
  "n" : 0,
  "nModified" : 0,
  "writeErrors" : [
    {
      "index" : 0,
      "code" : 121,
      "errmsg" : "Document failed validation"
    }
  ],
  "ok" : 1,
  "operationTime" : Timestamp(1234567890, 1)
}
```

moderate將驗證層級更新為允許成功更新上述文件：

```
db.runCommand({
  "collMod" : "employees",
  validationLevel : "moderate"
})

db.runCommand({
  update: "employees",
  updates: [{
    q: { "employeeId": "c721a" },
```

```

    u: { age: 25 , salary : 1000},
    upsert: true }]
  })

```

Amazon DocumentDB 返回以下輸出：

```

{
  "n" : 1,
  "nModified" : 1,
  "ok" : 1,
  "operationTime" : Timestamp(1234567890, 1)
}

```

使用 \$J 綱要檢索文檔

`$jsonSchema` 運算子可當做篩選器來查詢符合 JSON 結構描述的文件。這是一個頂級運算符，它可以出現在過濾器文檔中作為頂級字段或與查詢運算符（如 `$and`，`$or`，和 `$nor`）一起使用 `$nor`。下列範例會示範使用 `$JsonSchema` 做為個別篩選器和其他篩選運算子：

插入到一個「員工」集合的文檔：

```

{ "name" : { "firstName" : "Carol", "lastName" : "Smith" }, "employeeId" : "c720a",
  "salary" : 1000 }
{ "name" : { "firstName" : "Emily", "lastName" : "Brown" }, "employeeId" : "c720b",
  "age" : 25, "salary" : 1050.2 }
{ "name" : { "firstName" : "William", "lastName" : "Taylor" }, "employeeId" : "c721a",
  "age" : 24, "salary" : 1400.5 }
{ "name" : { "firstName" : "Jane", "lastName" : "Doe" }, "employeeId" : "c721a",
  "salary" : 1300 }

```

僅使用 `$jsonSchema` 運算符過濾的集合：

```

db.employees.find({
  $jsonSchema: { required: ["age"] } })

```

Amazon DocumentDB 返回以下輸出：

```

{ "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"), "name" : { "firstName" : "Emily",
  "lastName" : "Brown" }, "employeeId" : "c720b", "age" : 25, "salary" : 1050.2 }
{ "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"), "name" : { "firstName" : "William",
  "lastName" : "Taylor" }, "employeeId" : "c721a", "age" : 24, "salary" : 1400.5 }

```

使用\$jsonSchema運算子和另一個運算子篩選的集合：

```
db.employees.find({
  $or: [{ $jsonSchema: { required: ["age", "name"]}},
        { salary: { $lte:1000}}]}]);
```

Amazon DocumentDB 返回以下輸出：

```
{ "_id" : ObjectId("64e5f8886218c620cf0e8f8a"), "name" : { "firstName" : "Carol",
"lastName" : "Smith" }, "employeeId" : "c720a", "salary" : 1000 }
{ "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"), "name" : { "firstName" : "Emily",
"lastName" : "Brown" }, "employeeId" : "c720b", "age" : 25, "salary" : 1050.2 }
{ "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"), "name" : { "firstName" : "William",
"lastName" : "Taylor" }, "employeeId" : "c721a", "age" : 24, "salary" : 1400.5 }
```

使用\$jsonSchema運算符和聚合過濾器過濾的集合：\$match

```
db.employees.aggregate(
  [{ $match: {
    $jsonSchema: {
      required: ["name", "employeeId"],
      properties: {"salary" :{"bsonType": "double"}}
    }
  }
}]
)
```

Amazon DocumentDB 返回以下輸出：

```
{
  "_id" : ObjectId("64e5f8886218c620cf0e8f8a"),
  "name" : { "firstName" : "Carol", "lastName" : "Smith" },
  "employeeId" : "c720a",
  "salary" : 1000
}
{
  "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"),
  "name" : { "firstName" : "Emily", "lastName" : "Brown" },
  "employeeId" : "c720b",
  "age" : 25,
  "salary" : 1050.2
}
```



```
{
  "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"),
  "name" : { "firstName" : "William", "lastName" : "Taylor" },
  "employeeId" : "c721a",
  "age" : 24,
  "salary" : 1400.5
}
{
  "_id" : ObjectId("64e5f9786218c620cf0e8f8d"),
  "name" : { "firstName" : "Jane", "lastName" : "Doe" },
  "employeeId" : "c721a",
  "salary" : 1300
}
```

檢視現有的驗證規則

若要查看集合上現有的驗證規則，請使用：

```
db.runCommand({
  listCollections: 1,
  filter: { name: 'employees' }
})
```

Amazon DocumentDB 返回以下輸出：

```
{
  "waitedMS" : NumberLong(0),
  "cursor" : {
    "firstBatch" : [
      {
        "name" : "employees",
        "type" : "collection",
        "options" : {
          "autoIndexId" : true,
          "capped" : false,
          "validator" : {
            "$jsonSchema" : {
              "bsonType" : "object",
              "title" : "employee validation",
              "required" : [
                "name",
                "employeeId",
                "salary"
              ]
            }
          }
        }
      }
    ]
  }
}
```

```
    ],
    "properties" : {
      "name" : {
        "bsonType" : "object",
        "properties" : {
          "firstName" : {
            "bsonType" : [
              "string"
            ]
          },
          "lastName" : {
            "bsonType" : [
              "string"
            ]
          }
        },
        "additionalProperties" : false
      },
      "employeeId" : {
        "bsonType" : "string",
        "description" : "Unique Identifier for employee"
      },
      "salary" : {
        "bsonType" : "double"
      },
      "age" : {
        "bsonType" : "number"
      }
    },
    "additionalProperties" : true
  }
},
"validationLevel" : "moderate",
"validationAction" : "error"
},
"info" : {
  "readOnly" : false
},
"idIndex" : {
  "v" : 2,
  "key" : {
    "_id" : 1
  },
  "name" : "_id_",
```

```
        "ns" : "test.employees"
      }
    }
  ],
  "id" : NumberLong(0),
  "ns" : "test.$cmd.listCollections"
},
"ok" : 1,
"operationTime" : Timestamp(1692788937, 1)
}
```

支援關鍵字

create和collMod指令支援下列欄位：

- **Validator**— 支援 \$jsonSchema 運算子。
- **ValidationLevel**— 支援offstrict、和moderate值。
- **ValidationAction**— 支援error值。

該 \$jsonSchema 運算符支持以下關鍵字：

- additionalItems
- additionalProperties
- allOf
- anyOf
- bsonType
- dependencies
- description
- enum
- exclusiveMaximum
- exclusiveMinimum
- items
- maximum
- minimum
- maxItems

- `minItems`
- `maxLength`
- `minLength`
- `maxProperties`
- `minProperties`
- `multipleOf`
- `not`
- `oneOf`
- `pattern`
- `patternProperties`
- `properties`
- `required`
- `title`
- `type`
- `uniqueItems`

限制

下列限制適用於`$jsonSchema`驗證：

- 當使用`$out`彙總階段寫入集合的作業時，集合不會保留驗證規則。若要取得輸出集合中的有效文件，建議您在`$match`彙總階段之前使用彙總階段中的`$jsonSchema$out`篩選器。結構描述驗證可以在`$out`彙總階段之後重新套用至輸出集合。

```
db.foo.aggregate([{$match: {$jsonSchema: {...}}}, {$out: "bar"}]);  
db.runCommand("collMod": "bar", validator: {$jsonSchema: {...}})
```

- 當操作失敗驗證規則時，Amazon DocumentDB 會傳回錯誤「文件驗證失敗」。
- Amazon DocumentDB 不支持`bypassDocumentValidation`操作中的`db.runCommand`操作。
- 不支援 Amazon DocumentDB 彈性叢集。`$jsonSchema`

以複本集形式連接至 Amazon DocumentDB

當您針對 Amazon DocumentDB 進行開發時 (具備 MongoDB 兼容性) ，我們建議您以複本集形式連接至您的叢集，並使用驅動程式的內建讀取偏好設定功能，將讀取分配到複本執行個體。本節將深入探討其意義，並以做為範例，說明如何以複本集形式連接至您的 Amazon DocumentDB 叢集。

Amazon DocumentDB 有三個端點，可供您用來連接至您的叢集：

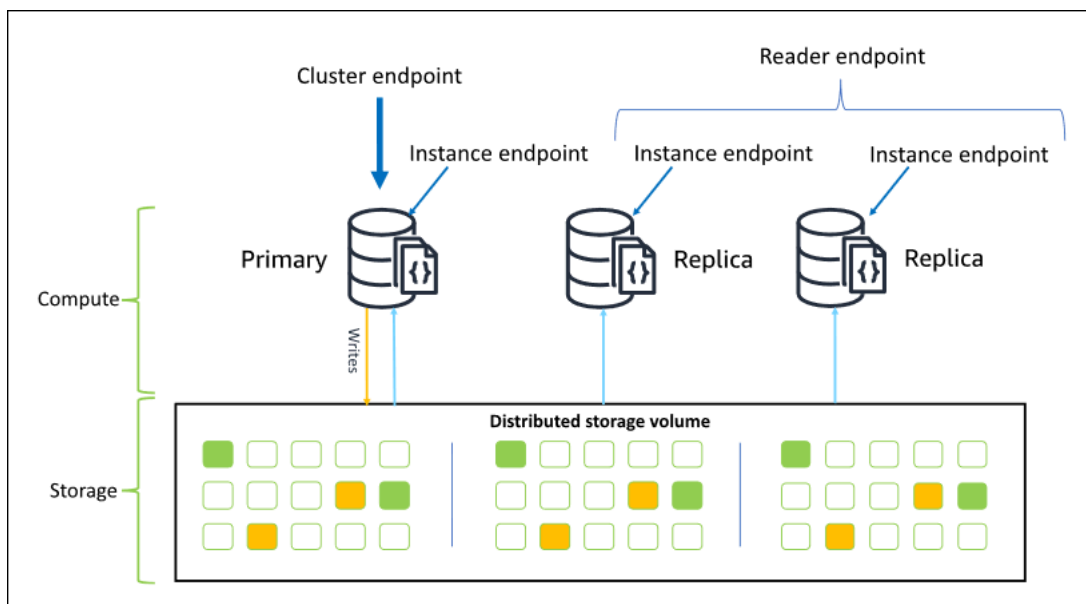
- 叢集端點
- 讀取器端點
- 執行個體端點

在大部分情況下，當您連接至 Amazon DocumentDB 時，我們建議您使用該叢集端點。這是將指向您叢集中主要執行個體的 CNAME，如下圖所示。

使用 SSH 通道時，建議您使用叢集端點連線至叢集，並且不要嘗試以複本集模式 (即在連線字串中指定 `replicaSet=rs0`) 進行連線，因為這會導致錯誤。

Note

如需 Amazon DocumentDB 端點的詳細資訊，請參[Amazon DocumentDB 端點](#)。



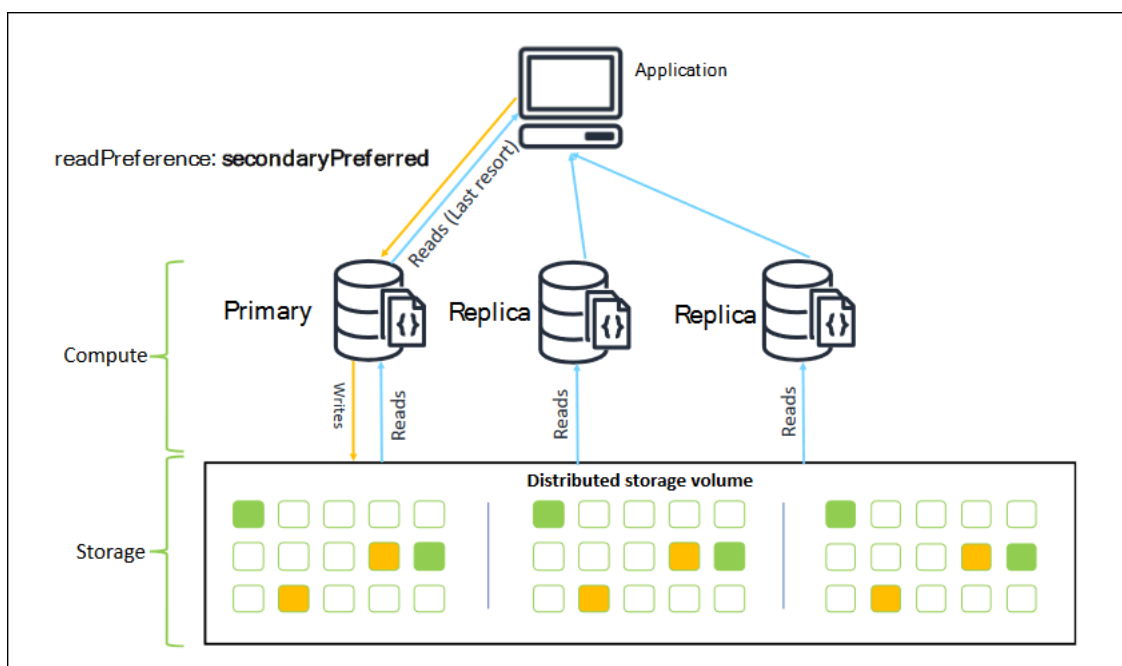
使用叢集端點時，您可以連接至處在複本集模式下的叢集。您就可以使用內建的讀取偏好設定驅動程式功能。在下列範例中，指定 `/?replicaSet=rs0` 時，系統會通知軟體開發套件您想要以複本集形式進行連接。如果您省略 `/?replicaSet=rs0`，用戶端會將所有請求路由到叢集端點，也就是您的主要執行個體。

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a
## replica set and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://<user-name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?replicaSet=rs0')
```

以複本集形式連接的優點，就是其可讓軟體開發套件自動探索叢集拓撲圖，包括當叢集新增或移除執行個體時。然後，您就可以將讀取請求路由至複本執行個體，提高叢集的使用效率。

當您以複本集形式連接時，您可以指定 `readPreference` 進行連線。如果您指定的是 `secondaryPreferred` 的讀取偏好設定，則用戶端會將讀取查詢路由到您的複本，並將查詢寫入到您的主要執行個體 (如下圖所示)。這是較佳的叢集資源使用方式。如需詳細資訊，請參閱 [讀取偏好設定選項](#)。

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a
## replica set and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://<user-name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?replicaSet=rs0&readPreference=secondaryPreferred')
```



來自 Amazon DocumentDB 複本的讀取最終會保持一致。它們會依據主要執行個體上的相同寫入順序來傳回資料，而且複寫延遲通常少於 50 毫秒。您可以使用 Amazon CloudWatch 指標來監控叢集的複本延遲 `DBInstanceReplicaLag` 和 `DBClusterReplicaLagMaximum`。如需詳細資訊，請參閱 [使用監控亞馬遜文檔數據庫 CloudWatch](#)。

不同於傳統的整合型資料庫架構，Amazon DocumentDB 會分開進行儲存和運算。假設您使用這種現代架構，那麼我們鼓勵您讀取複本執行個體上的擴展。複本執行個體上的讀取不會封鎖從主要執行個體複寫的寫入。您在叢集中最多可以新增 15 個讀取複本執行個體，並將讀取量擴展到每秒數百萬次讀取。

以複本集形式連接，並將讀取分配至複本的主要優點，在於其會在叢集中增加可供您應用程式使用的整體資源。我們建議以複本集形式連接來進行最佳實務。此外，我們最常在下列案例中建議使用這個做法：

- 您正使用主要執行個體上將近 100% 的 CPU。
- 緩衝區快取命中率接近零。
- 您的個別執行個體的連線或游標已達上限。

您可以選擇向上擴展叢集執行個體大小，而且在某些情況下，這可能是擴展叢集的最佳方式。但是，您也應該考慮如何提高叢集中既有複本的使用量。這可讓您直接增加擴展，而不用增加使用更大執行個體類型的成本。我們也建議您監控和提醒這些限制 (也就是 `CPUUtilization`、`DatabaseConnections`，以及 `BufferCacheHitRatio`)，以便您能掌握資源何時受到大量使用。

如需詳細資訊，請參閱下列主題：

- [Amazon DocumentDB 的最佳實踐](#)
- [Amazon DocumentDB 配額和限制](#)

使用叢集連線

請考慮使用叢集中所有連線的案例。例如，`r5.2xlarge` 執行個體的連線數目限為 4,500 個 (而開放游標限制為 450 個)。如果您建立三個執行個體 Amazon DocumentDB 叢集，並使用叢集端點僅連接至主要執行個體，則進行開放連線和遊標的叢集分別限制為 4,500 個和 450 個。如果您建立的應用程式會使用多個將從容器啟動的 Worker，那麼您可能會達到這些限制。這些容器會一次開啟多個連線，並使叢集達到飽和。

反之，您可以複本集形式連接至 Amazon DocumentDB 叢集，並將讀取分配到複本執行個體。接著，您可以將叢集中的可用連線和游標數量成功增為三倍，分別為 13,500 和 1,350。將更多執行個體新增至叢集時，只會增加讀取工作負載的連線數和游標數。如果您需要在叢集中增加寫入連線數目，建議您增加執行個體大小。

Note

large、xlarge 和 2xlarge 執行個體的連線數目會隨著執行個體大小增加，最多達 4,500 個。4xlarge 個或超過 4,500 個執行個體時，每個執行個體的連線數目上限。如需依執行個體類型限制的詳細資訊，請參閱 [執行個體限制](#)。

一般而言，我們不建議您使用 secondary 的讀取偏好設定來連接至叢集。原因在於如果您的叢集中沒有複本執行個體，該讀取將會失敗。例如，假設您擁有兩個執行個體的 Amazon DocumentDB 叢集，其中有一個主要執行個體和一個複本。如果複本發生問題，則設為 secondary 之連線集區的讀取請求將會失敗。使用 secondaryPreferred 的優點，在於如果用戶端找不到適合的複本執行個體來進行連接，則其會回復到主要執行個體來進行讀取。

多個連線集區

在某些情況下，應用程式中的讀取必須具備先寫後讀一致性，而這只能透過 Amazon DocumentDB 的主要執行個體提供服務。在這些情況下，您可以建立兩個用戶端連線集區：一個用於寫入，另一個用於需要先寫後讀一致性的讀取。若要這樣做，您的程式碼應該與下列類似。

```
## Create a MongoDB client,
##   open a connection to Amazon DocumentDB as a replica set and specify the
##   readPreference as primary
clientPrimary = pymongo.MongoClient('mongodb://<user-
name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=primary')

## Create a MongoDB client,
##   open a connection to Amazon DocumentDB as a replica set and specify the
##   readPreference as secondaryPreferred
secondaryPreferred = pymongo.MongoClient('mongodb://<user-
name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=secondaryPreferred')
```

另一種選擇是建立單一連線集區，並覆寫指定集合的讀取偏好設定。


```
##Specify the collection and set the read preference level for that collection
col = db.review.with_options(read_preference=ReadPreference.SECONDARY_PREFERRED)
```

總結

若要提高叢集資源的使用效能，建議您使用複本集模式連接至您的叢集。如果您的應用程式適合這種做法，您可以將讀取分配到該複本執行個體，讀取擴展您的應用程式。

從亞馬遜 VPC 外部連接到 Amazon DocumentDB 集群

Amazon DocumentDB 叢集中部署 Amazon Virtual Private Cloud VPC。它們可以由 Amazon EC2 執行個體或部署在相同 Amazon VPC 中的其他 AWS 服務直接存取。此外，可透過 VPC 對等互連，由 EC2 執行個體或相同或其他區域中不同 AWS 區域 VPC 中的其他 AWS 服務存取 Amazon DocumentDB。

但是，假設您的使用案例需要您 (或您的應用程式) 從叢集的 VPC 外部存取 Amazon DocumentDB 資源。在這種情況下，您可以使用 SSH 通道 (也稱為連接埠轉送) 來存取您的 Amazon DocumentDB 資源。

不過，深入討論 SSH 通道超出本主題的範圍。如需 SSH 通道的詳細資訊，請參閱以下內容：

- [SSH 通道](#)
- [SSH 連接埠轉送範例](#)，具體來說是 [本機轉送](#) 一節

要建立 SSH 執行個體執行，您需要使用與 Amazon DocumentDB 叢集相同的 Amazon VPC 中執行的 Amazon EC2 執行個體執行。您可以使用相同 VPC 中現有的 EC2 執行個體做為您的叢集，或建立一個叢集。如需詳細資訊，請參閱適用您作業系統的主題：

- [Amazon EC2 入](#)
- [Amazon EC2 入](#)

通常您可能會使用下列命令連接到 EC2 執行個體。

```
ssh -i "ec2Access.pem" ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com
```

如果是這樣，您可以在本機電腦上執行下列命令，以設定連至 Amazon DocumentDB 叢集的 SSHsample-cluster.node.us-east-1.docdb.amazonaws.com 通道。-L 旗標用於轉送本機

連接埠。使用 SSH 通道時，建議您使用叢集端點連線至叢集，並且不要嘗試以複本集模式 (即在連線字串中指定 `replicaSet=rs0`) 進行連線，因為這會導致錯誤。

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-east-1.docdb.amazonaws.com:27017
ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -N
```

建立 SSH 通道之後，您發出的所有命令 `localhost:27017` 都會轉送到在 Amazon VPC 中 `sample-cluster` 執行的 Amazon DocumentDB 叢集。如果您的 Amazon DocumentDB 叢集上已啟用傳輸層安全性 (TLS)，您需要從下載 Amazon DocumentDB 的公開金鑰 <https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem>。下列作業會下載此檔案：

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Note

在預設情況下，對於 Amazon DocumentDB 叢集，預設啟用 TLS。不過，您可以停用它。如需詳細資訊，請參閱 [管理 Amazon DocumentDB 叢集 TLS 設定](#)。

若要從 Amazon VPC 外部連線到 Amazon 文件資料庫叢集，請使用下列命令。

```
mongo --sslAllowInvalidHostnames --ssl --sslCAFile global-bundle.pem --username
<yourUsername> --password <yourPassword>
```

從工作室 3T 連接到 Amazon DocumentDB 集群

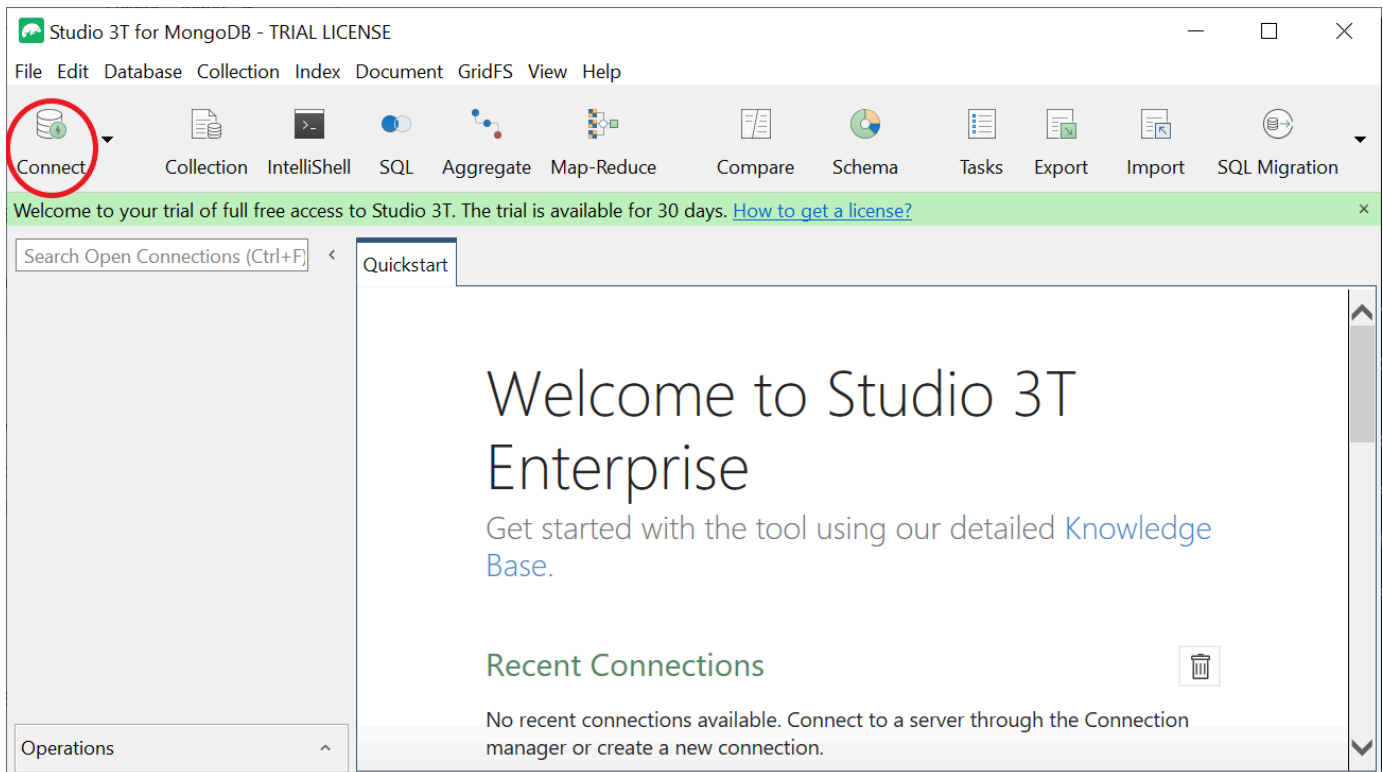
[工作室 3T](#) 是一個流行的圖形用戶界面和 IDE 開發人員和數據工程師誰與 MongoDB 的工作。它提供了數種強大的功能樹狀結構、資料表和 JSON 檢視、CSV、JSON、SQL 和 BSONGODUMP 中的輕鬆匯入/匯出、彈性的查詢選項、視覺化 drag-and-drop 使用者介面、具有自動完成功能的內建蒙戈殼層、彙總管線編輯器以及 SQL 查詢支援。

先決條件

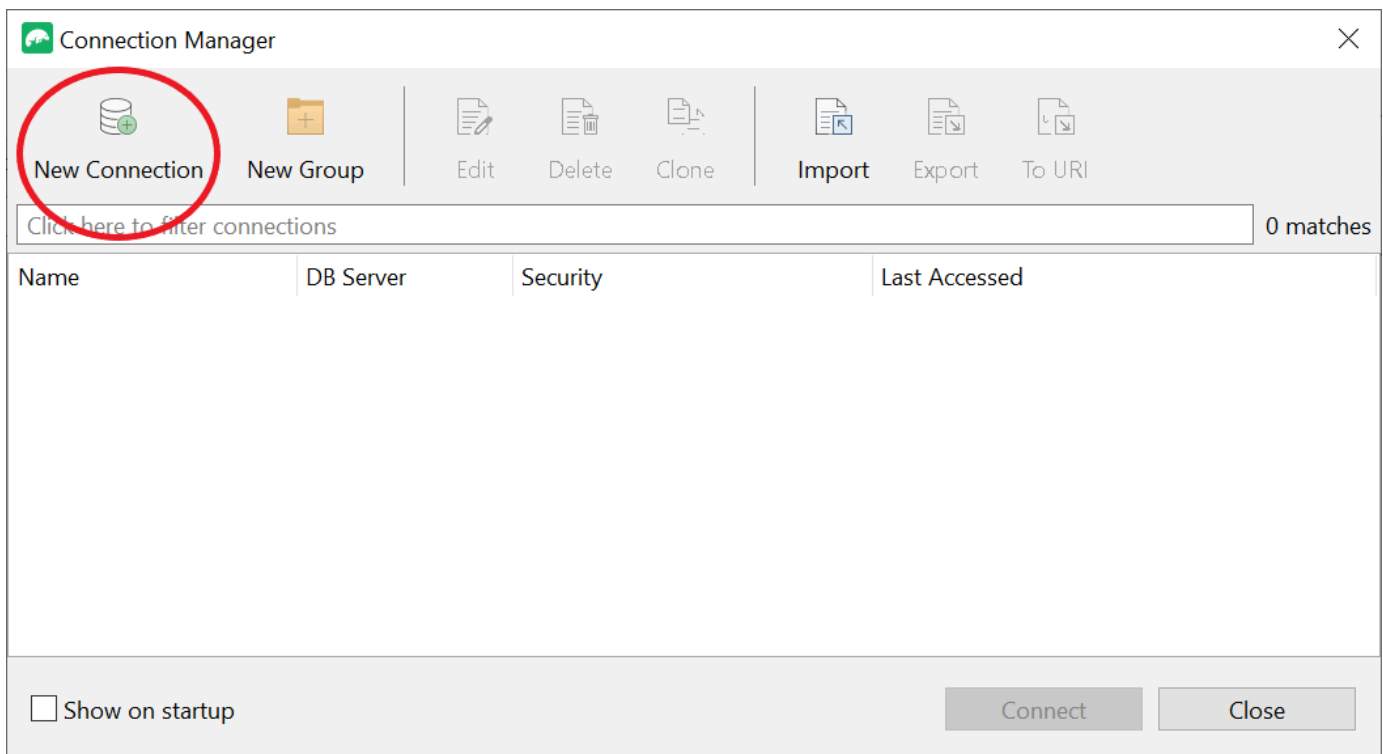
- 如果您還沒有在亞馬遜 EC2 上執行的亞馬遜 DocumentDB 叢集，請按照有關如何[與亞 Amazon EC2 Connect](#) 的說明進行操作。
- 如果您沒有工作室 3T，請[下載並安裝它](#)。

與工作室 3T Connect

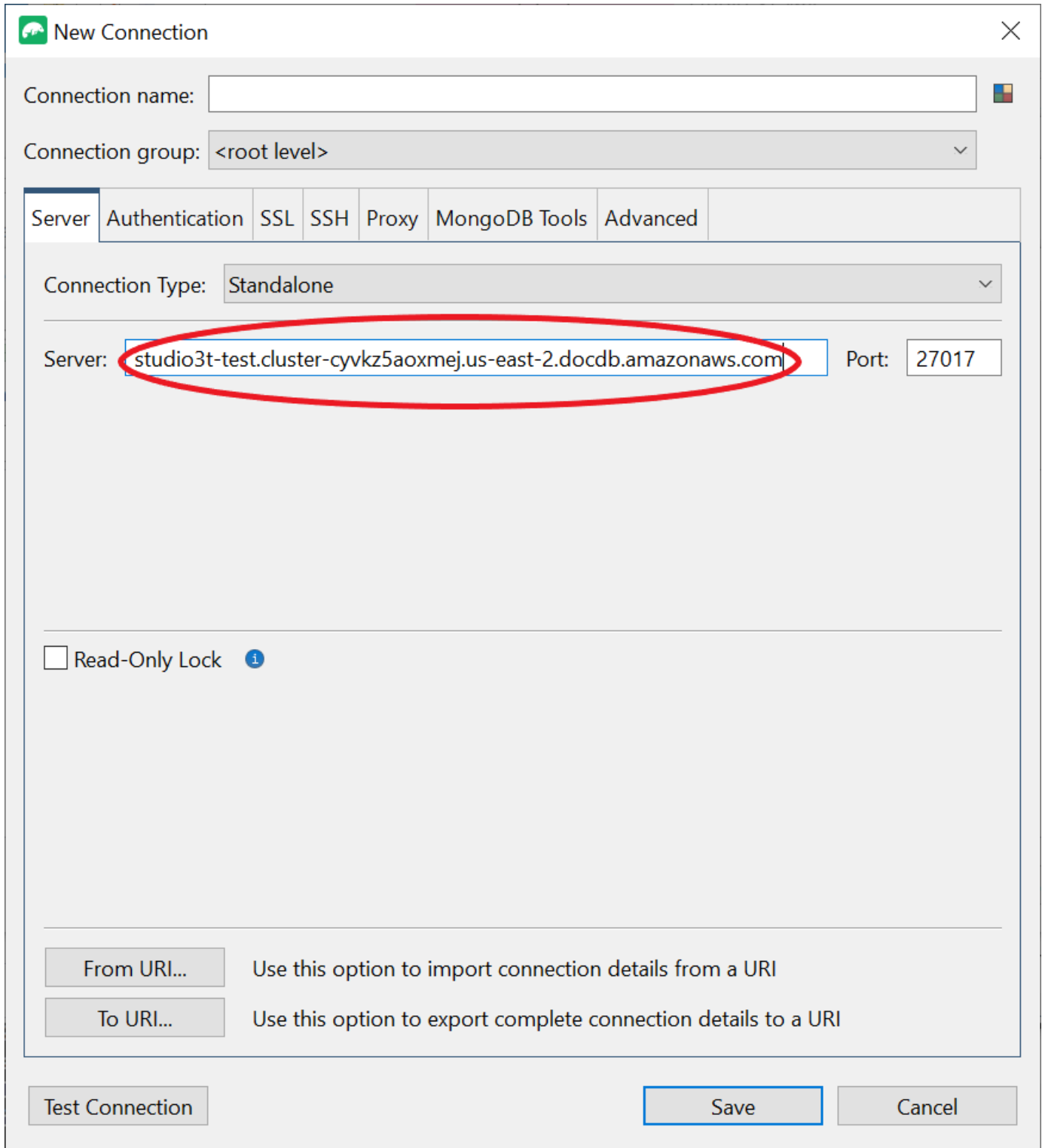
1. 選擇工具列左上角的 [Connect]。



2. 選擇工具列左上角的 [新增連線]。



3. 在「伺服器」索引標籤的「伺服器」欄位中，輸入叢集端點資訊。



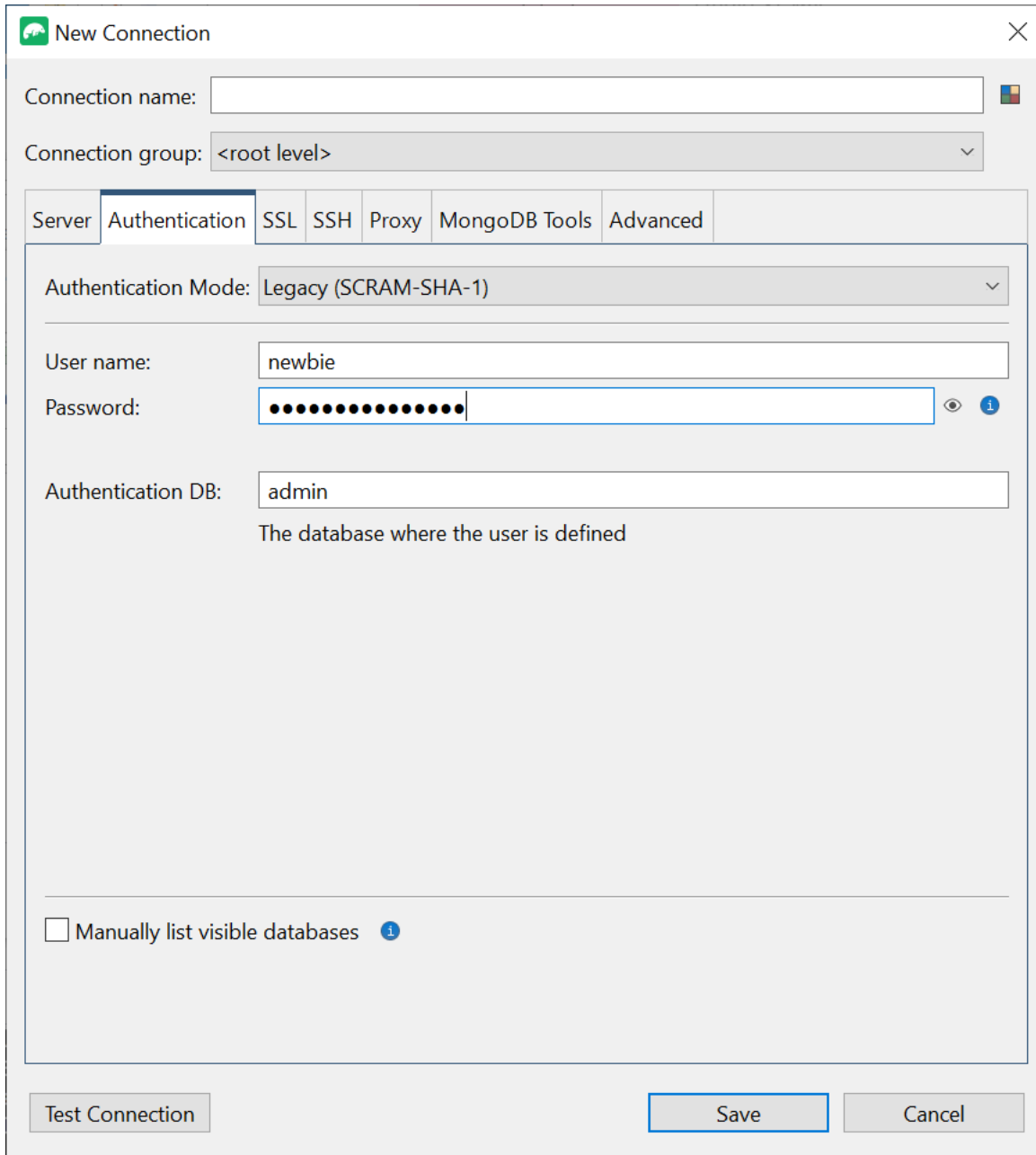
The screenshot shows the 'New Connection' dialog box with the following fields and options:

- Connection name:
- Connection group: <root level>
- Server: (highlighted with a red oval)
- Port:
- Read-Only Lock:
- From URI...: Use this option to import connection details from a URI
- To URI...: Use this option to export complete connection details to a URI
- Buttons: Test Connection, Save, Cancel

Note

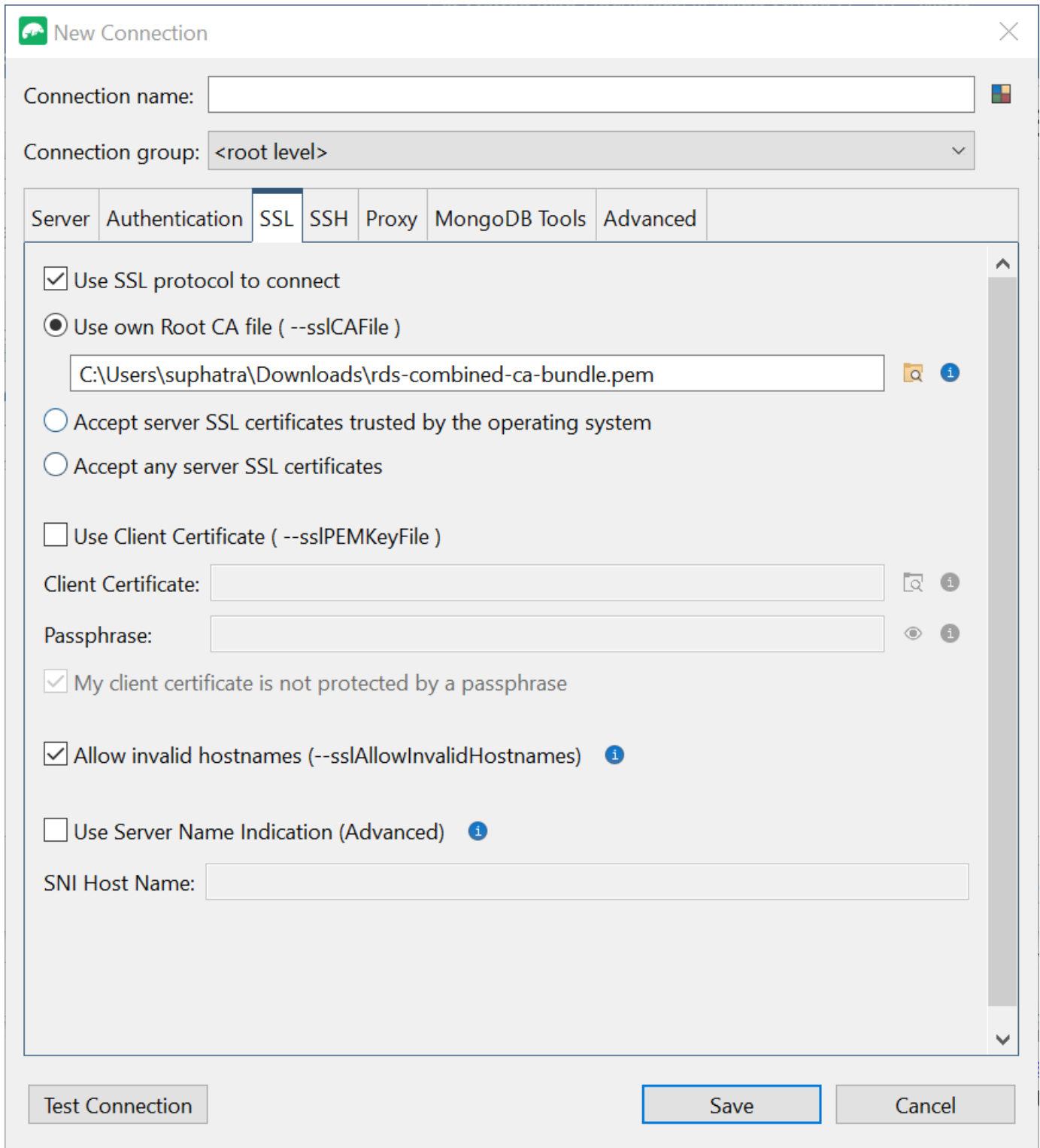
找不到您的叢集端點？只需按照[這裡](#)的步驟。

4. 選擇驗證索引標籤，然後在驗證模式的下拉式功能表中選取舊版。



The screenshot shows the 'New Connection' dialog box with the 'Authentication' tab selected. The 'Authentication Mode' dropdown is set to 'Legacy (SCRAM-SHA-1)'. The 'User name' field contains 'newbie'. The 'Password' field is masked with 12 dots. The 'Authentication DB' field contains 'admin'. Below the 'Authentication DB' field, there is a note: 'The database where the user is defined'. At the bottom, there is a checkbox for 'Manually list visible databases' which is unchecked. The 'Save' button is highlighted with a blue border.

5. 在使用者名稱和密碼欄位中輸入您的使用者名稱和憑證。
6. 選擇 SSL 選項卡，然後選中使用 SSL 協議進行連接。



New Connection

Connection name:

Connection group: <root level>

Server | Authentication | **SSL** | SSH | Proxy | MongoDB Tools | Advanced

Use SSL protocol to connect

Use own Root CA file (--sslCAFile)

Accept server SSL certificates trusted by the operating system

Accept any server SSL certificates

Use Client Certificate (--sslPEMKeyFile)

Client Certificate:

Passphrase:

My client certificate is not protected by a passphrase

Allow invalid hostnames (--sslAllowInvalidHostnames)

Use Server Name Indication (Advanced)

SNI Host Name:

Test Connection Save Cancel

7. 選擇「使用自己的根 CA 檔案」。然後新增 Amazon DocumentDB 證 (如果您的 DocumentDB 叢集上已停用 SSL，您可以略過此步驟)。核取此方塊可允許無效的主機名稱。

New Connection

Connection name:

Connection group: <root level>

Server Authentication **SSL** SSH Proxy MongoDB Tools Advanced

Use SSL protocol to connect

Use own Root CA file (--sslCAFile)

Accept server SSL certificates trusted by the operating system

Accept any server SSL certificates

Use Client Certificate (--sslPEMKeyFile)

Client Certificate:

Passphrase:

My client certificate is not protected by a passphrase

Allow invalid hostnames (--sslAllowInvalidHostnames)

Use Server Name Indication (Advanced)

SNI Host Name:

Test Connection Save Cancel

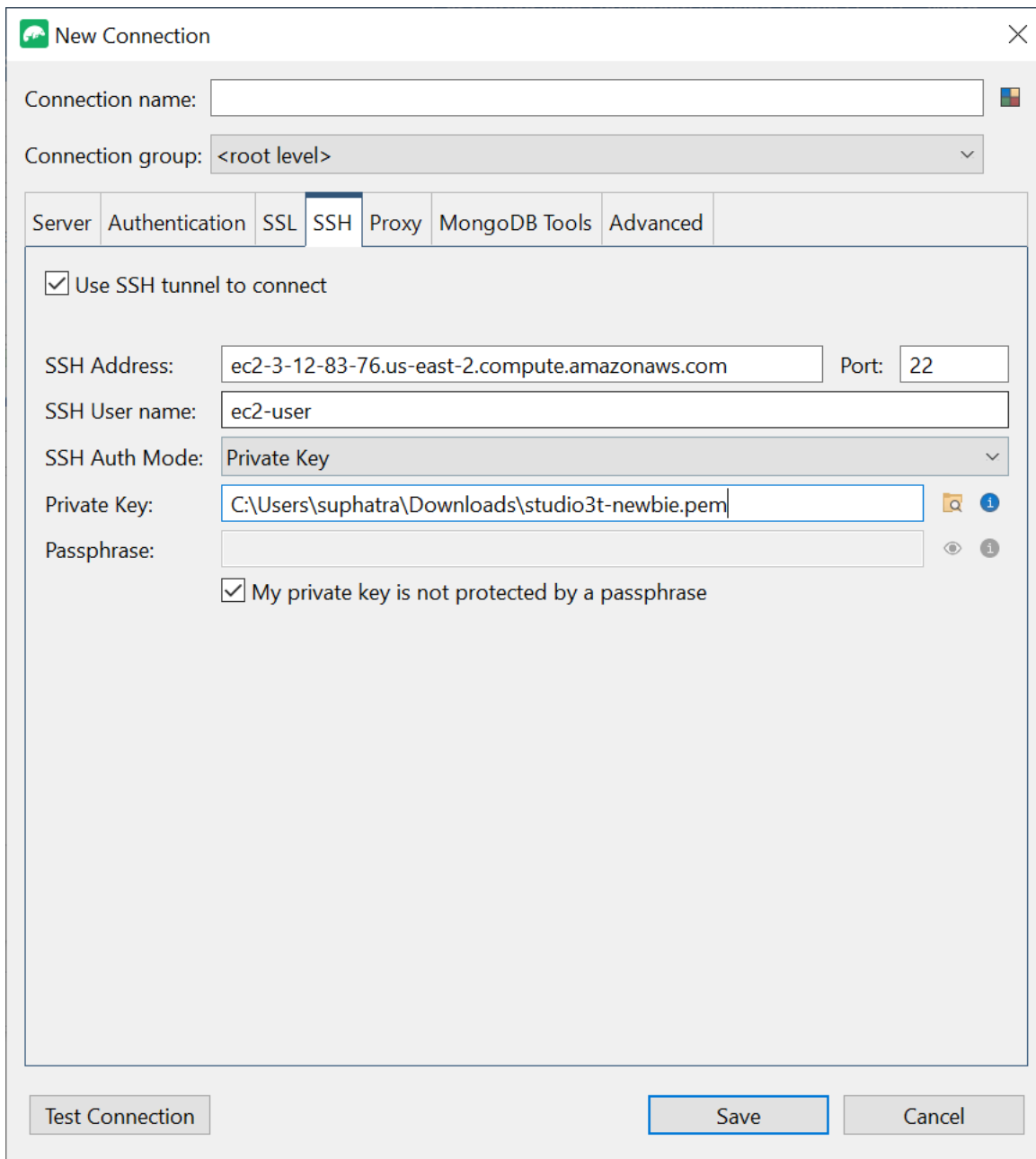
Note

沒有證書嗎？您可以使用以下命令下載它：

```
wget https://truststore.pki.rds.amazonaws.com/global/global-  
bundle.pem
```

8. 如果您是從 Amazon VPC 以外的用戶端機器連線，則需要建立 SSH 通道。您將在 SSH 選項卡中執行此操作。
 - a. 勾選 [使用 SSH 通道] 方塊，然後在 [SSH 位址] 欄位中輸入 SSH 位址。這是您的執行個體公用 DNS (IPV4)。您可以從 [Amazon EC2 管理主控台](#) 取得此 URL。
 - b. 輸入您的用戶名。這是您的 Amazon EC2 實例的用戶名
 - c. 針對 SSH 驗證模式，選取私密金鑰。在私密金鑰欄位中，選擇檔案尋找器圖示以尋找並選擇 Amazon EC2 執行個體的私密金鑰。這是您在 Amazon EC2 主控台中建立執行個體時儲存的 .pem 檔案 (key pair)。
 - d. 如果您使用的是 Linux/macOS 用戶端機器，則可能需要使用以下命令更改私鑰的權限：

```
chmod 400 /fullPathToYourPemFile/<yourKey>.pem
```

The screenshot shows the 'New Connection' dialog box with the 'SSH' tab selected. The configuration is as follows:

- Connection name: (empty text box)
- Connection group: <root level> (dropdown menu)
- SSH tab is active, with other tabs: Server, Authentication, SSL, Proxy, MongoDB Tools, Advanced.
- Use SSH tunnel to connect
- SSH Address: ec2-3-12-83-76.us-east-2.compute.amazonaws.com
- Port: 22
- SSH User name: ec2-user
- SSH Auth Mode: Private Key (dropdown menu)
- Private Key: C:\Users\suphatra\Downloads\studio3t-newbie.pem
- Passphrase: (empty text box)
- My private key is not protected by a passphrase

Buttons at the bottom: Test Connection, Save, Cancel.

Note

此 Amazon EC2 執行個體應與您的 DocumentDB 叢集位於相同的 Amazon VPC 和安全群組中。您可以從 [Amazon EC2 管理主控台](#) 取得 SSH 位址、使用者名稱和私密金鑰。

9. 現在，通過選擇測試連接按鈕測試您的配置。

New Connection

Connection name:

Connection group: <root level>

Server Authentication SSL SSH Proxy MongoDB Tools Advanced

Connection Type: Standalone

Server: robo3t-test.cluster-cyvkz5aoxmej.us-east-2.docdb.amazonaws.com Port: 27017

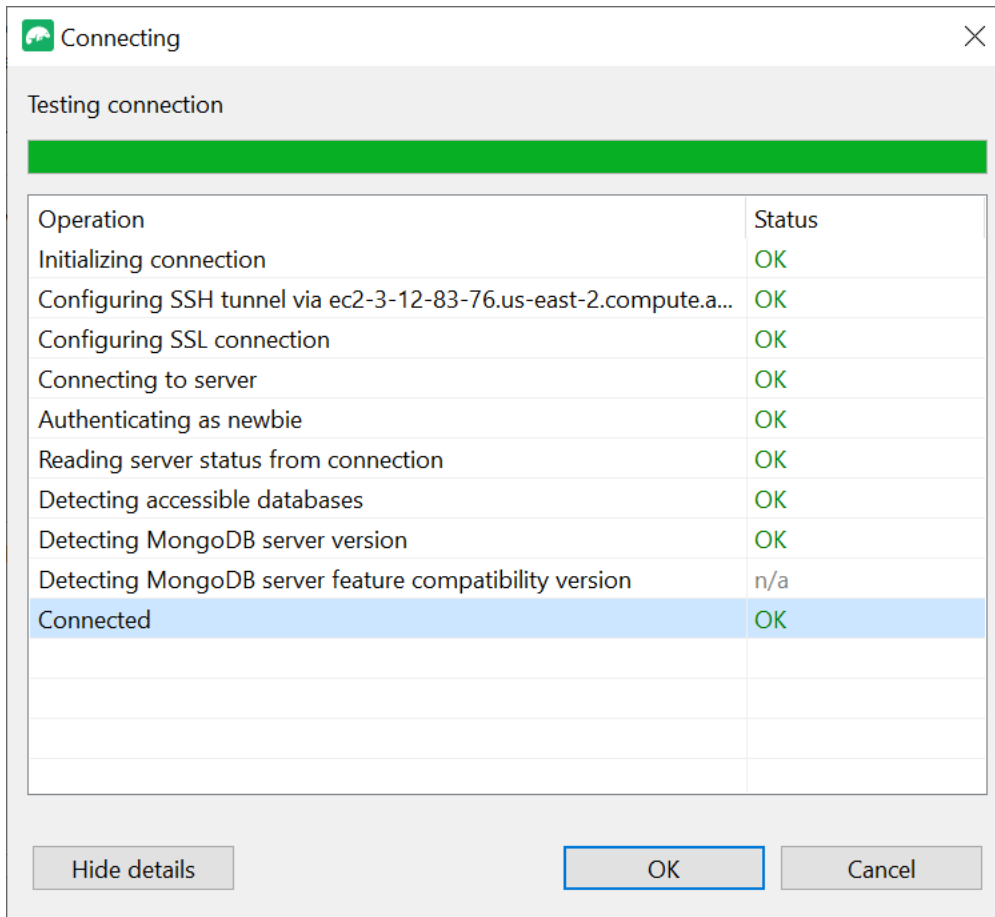
Read-Only Lock ⓘ

From URI... Use this option to import connection details from a URI

To URI... Use this option to export complete connection details to a URI

Test Connection Save Cancel

10. 診斷視窗應載入綠色列，表示測試成功。現在選擇 OK 關閉診斷窗口。



11. 選擇 [儲存] 儲存連線以供 future 使用。

New Connection

Connection name:

Connection group: <root level>

Server Authentication SSL SSH Proxy MongoDB Tools Advanced

Connection Type: Standalone

Server: robo3t-test.cluster-cyvkz5aoxmej.us-east-2.docdb.amazonaws.com Port: 27017

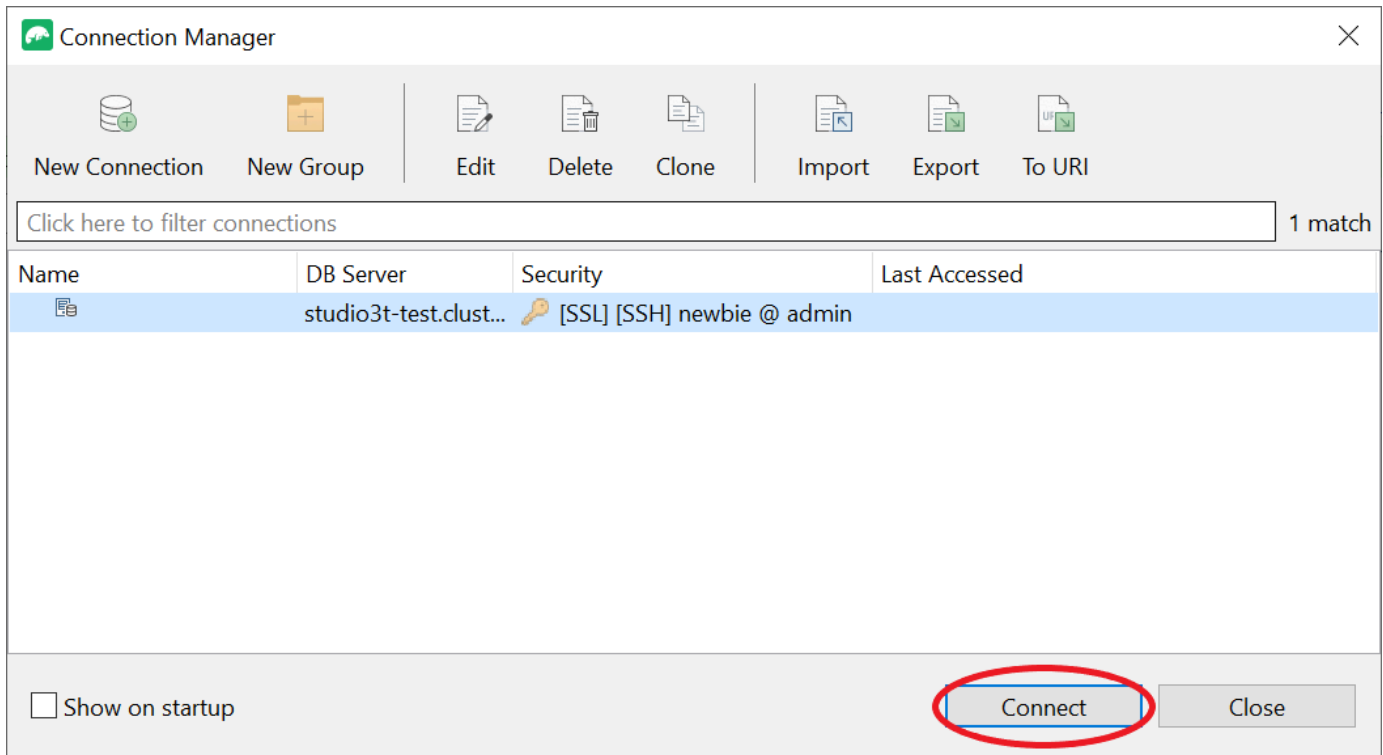
Read-Only Lock ?

From URI... Use this option to import connection details from a URI

To URI... Use this option to export complete connection details to a URI

Test Connection Save Cancel

12. 現在，選取您的叢集並選擇 [Connect]。



恭喜您！您現在已成功透過工作室 3T 連線到您的 Amazon DocumentDB 叢集。

使用 Connect 到 Amazon DocumentDB DataGrip

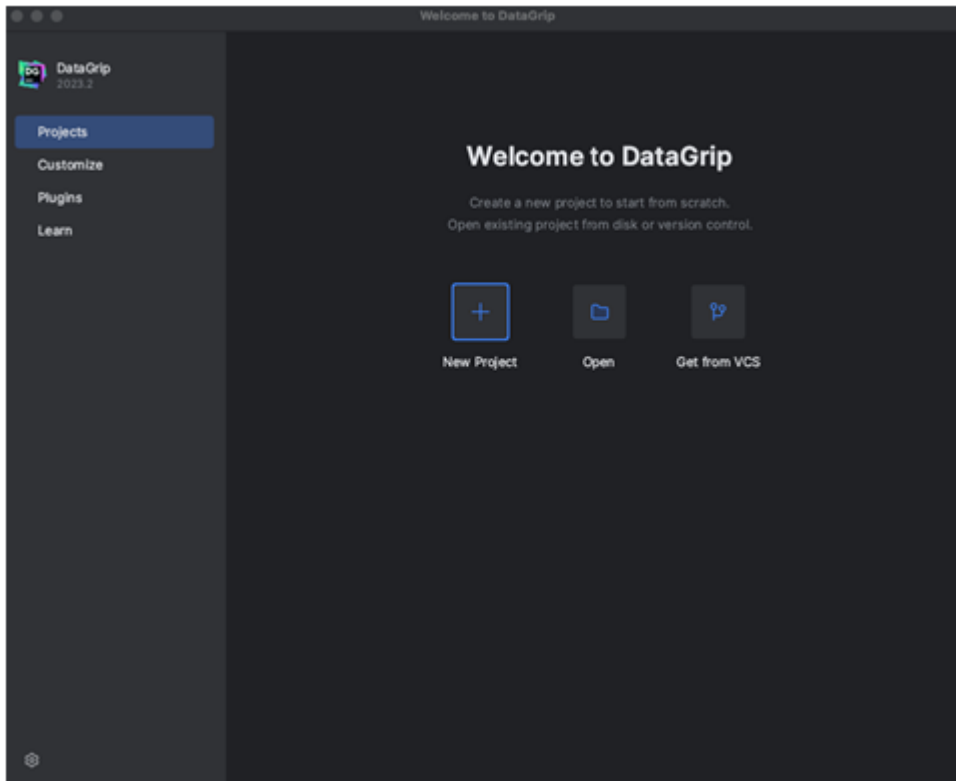
[DataGrip](#) 是一個功能強大的整合式開發環境 (IDE)，可支援各種資料庫系統，包括 Amazon DocumentDB。本節將引導您完成使用連接 Amazon DocumentDB 叢集的步驟 DataGrip，讓您可以使用圖形介面輕鬆管理和查詢資料。

先決條件

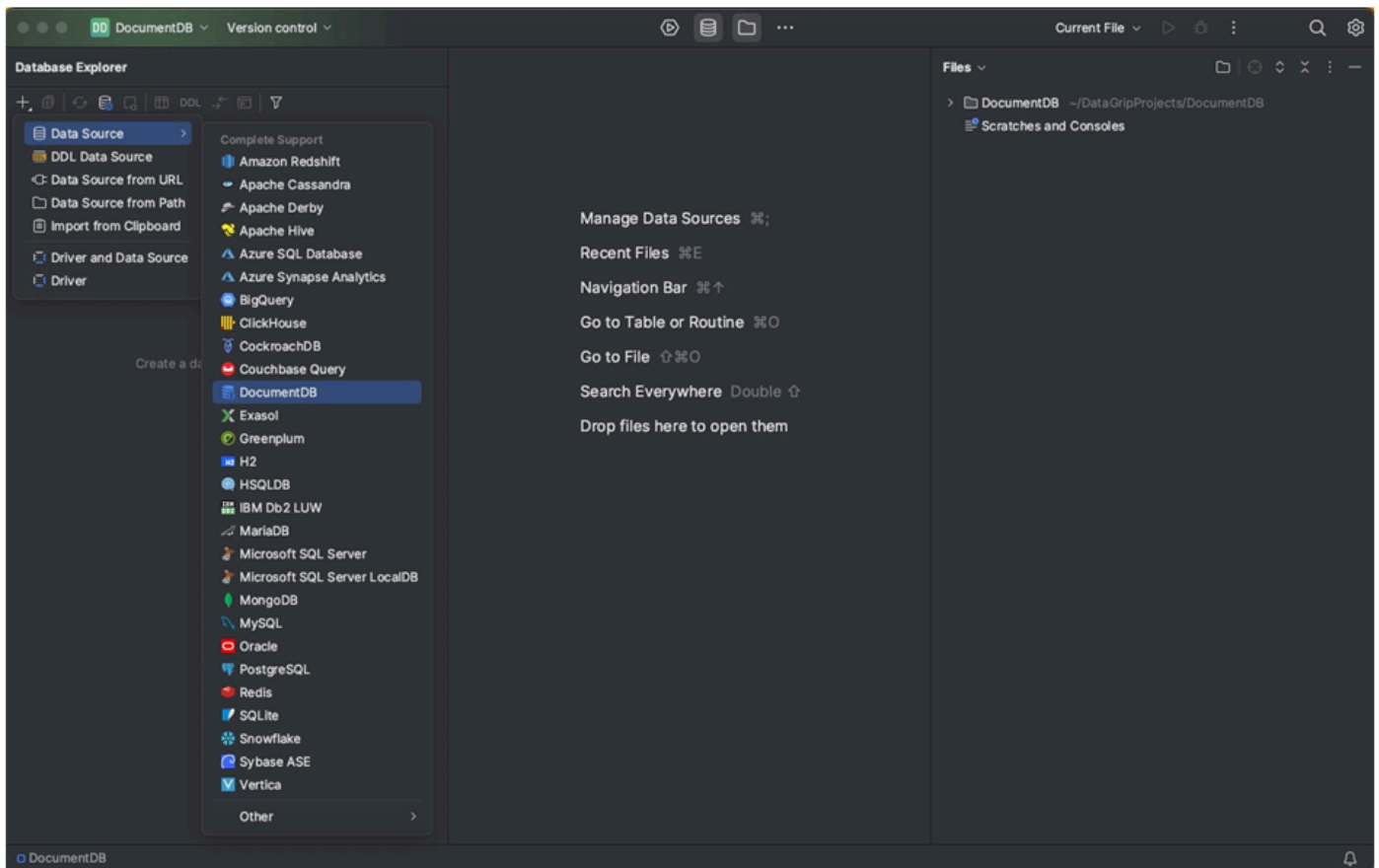
- DataGrip IDE 安裝在您的機器上。您可以從下載它 [JetBrains](#)。
- 在與您的 Amazon 文件資料庫叢集相同 VPC 中執行的 Amazon EC2 執行個體。您將使用此執行個體建立從本機電腦到 Amazon 文件資料庫叢集的安全通道。按照如何操作的說明進行操作 [使用 Amazon EC2 Connect](#)。
- 替代 Amazon EC2 執行個體、VPN 連線，或者您已經使用安全 VPN 存取 AWS 基礎設施。如果您偏好使用此選項，請依照指示 [使用安全地存取 Amazon DocumentDB](#)。AWS Client VPN

Connect 使用 DataGrip

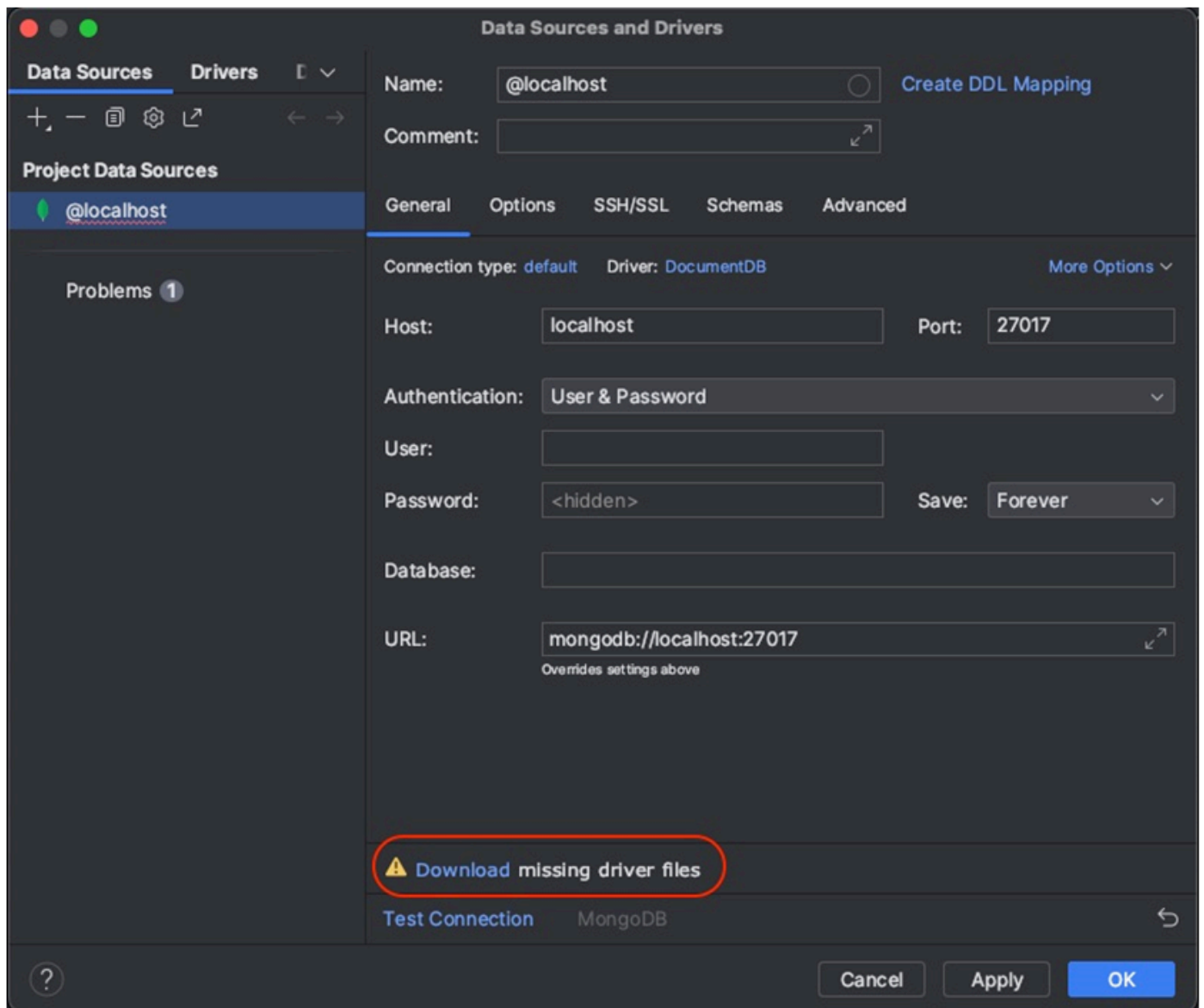
1. 在計算機 DataGrip 上啟動並創建一個新項目。



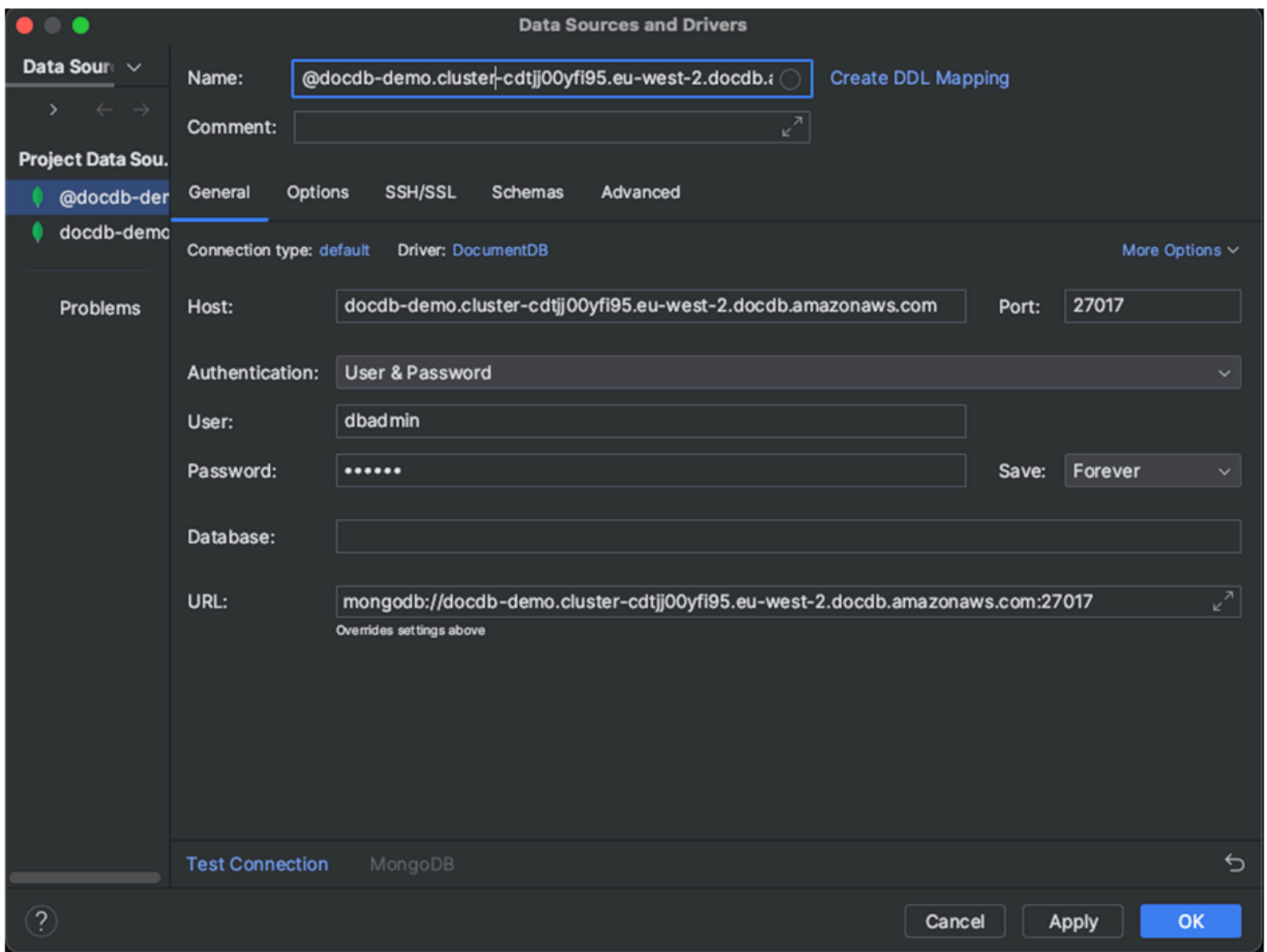
2. 使用下列其中一種方式新增資料來源：
 - a. 從主菜單中，導航到文件-新建-數據源，然後選擇 Document DB
 - b. 在「資料庫總管」中，按一下工具列中的新增圖示 (+)。瀏覽至資料來源，然後選取 DocumentDB。



3. 在 [一般] 索引標籤的 [資料來源] 頁面上，檢查連線設定區域底部是否有 [下載遺失的驅動程式檔案] 連結。按一下此連結可下載與資料庫互動所需的驅動程式。如需直接下載連結，請參閱 [JetBrains JDBC 驅動程式](#)。



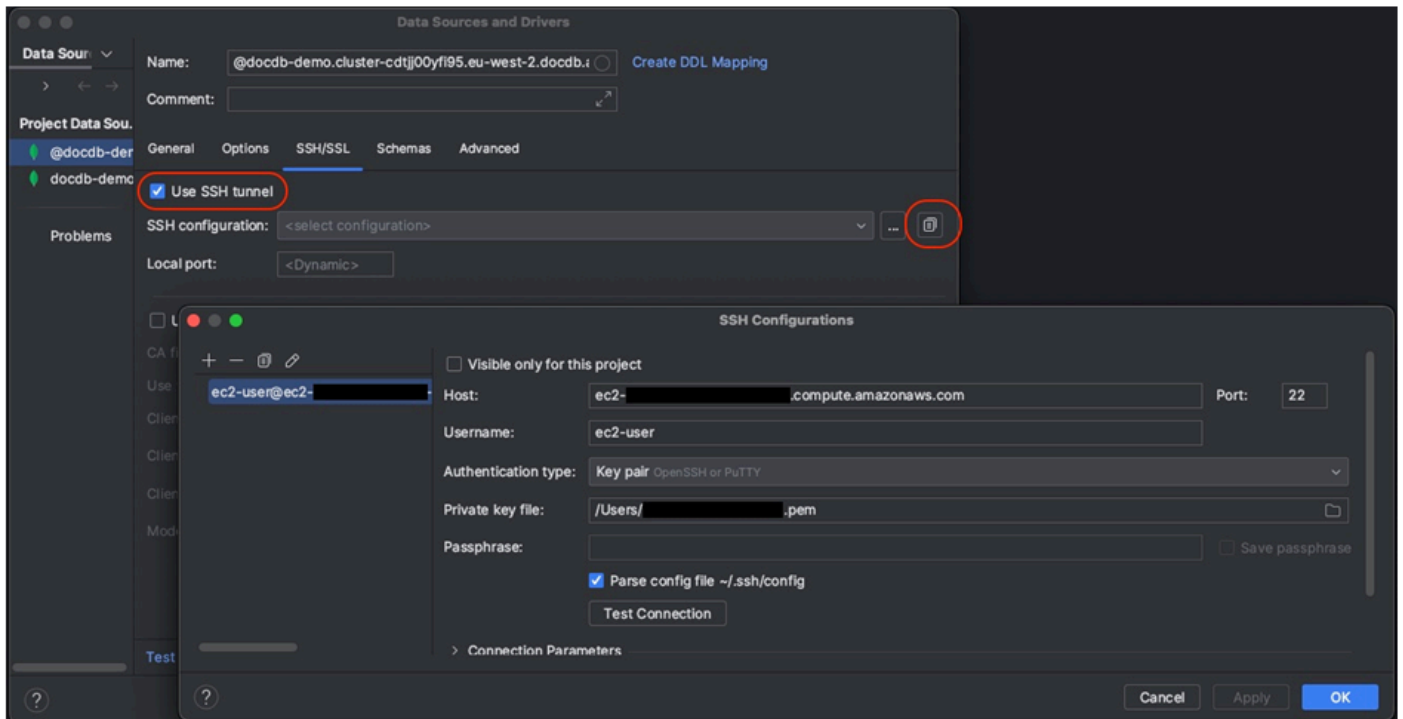
4. 在「一般」索引標籤中，指定連線詳細資訊：
 - a. 在「主機」欄位中，指定 Amazon DocumentDB 叢集端點。
 - b. 連接埠已設定為 27017。如果您的叢集部署在不同的連接埠上，請加以變更。
 - c. 對於驗證，選擇使用者和密碼。
 - d. 輸入您的使用者名稱和密碼資訊。
 - e. 「資料庫」欄位是選擇性的。您可以指定要連線的資料庫。
 - f. 當您新增上述詳細資訊時，URL 欄位會自動完成。



5. 在 SSH/SSL 選項卡中，啟用使用 SSH 通道，然後單擊圖標以打開 SSH 配置對話框。輸入下列資訊：
 - a. 在「主機」欄位中，輸入 Amazon EC2 執行個體的主機名稱。
 - b. 輸入 Amazon EC2 執行個體的使用者名稱和密碼。
 - c. 針對 [驗證類型] 選擇 [金鑰配對]。
 - d. 輸入您的私鑰文件。

Note

如果您使用的是 VPN 選項，則不需要設定 SSH 通道。



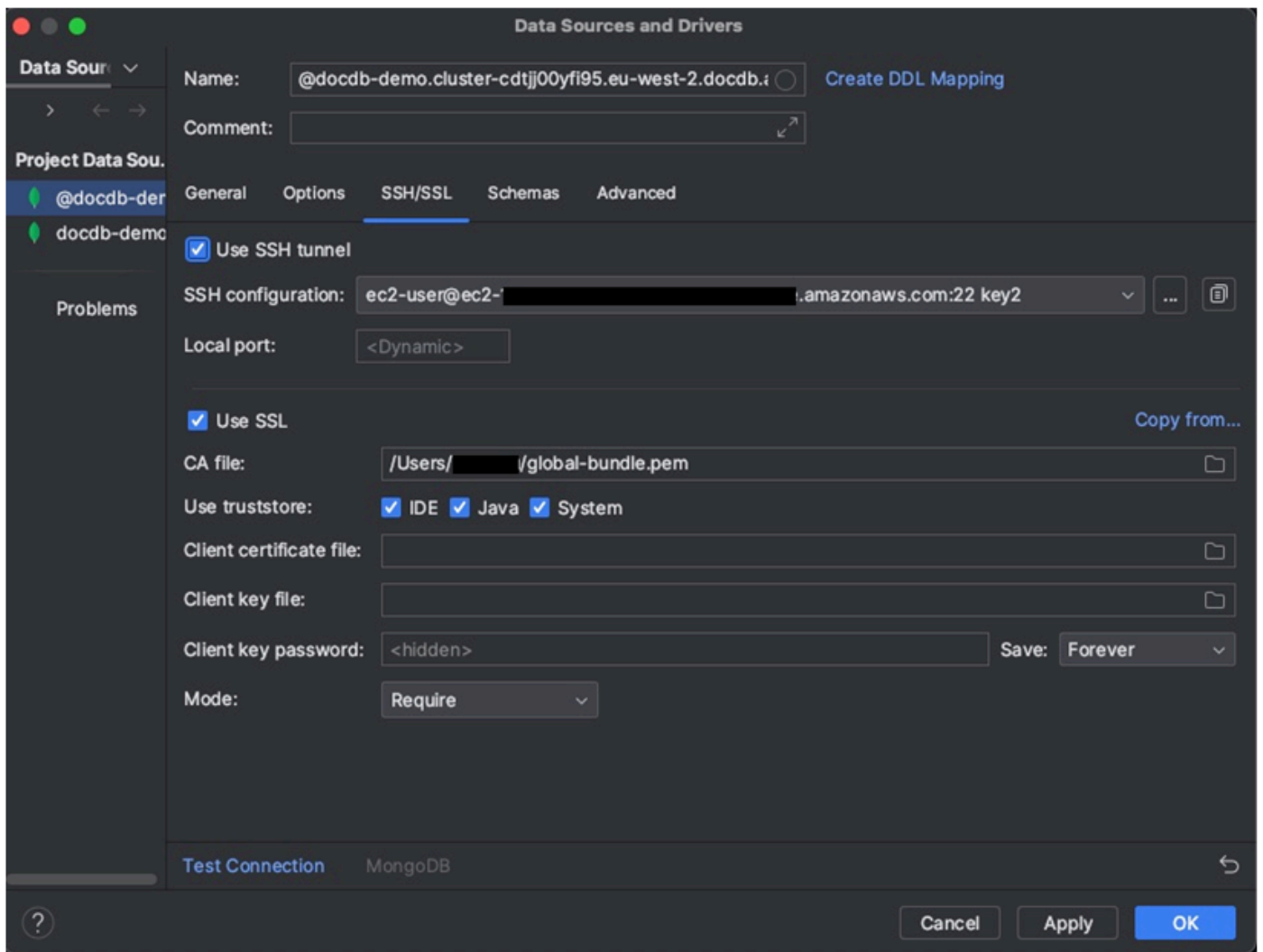
- 在 SSH/SSL 索引標籤中，啟用使用 SSL。在「CA 檔案」欄位中，輸入電腦上 `global-bundle.pem` 檔案的位置。對於「模式」，保留「需要」選項。

Note

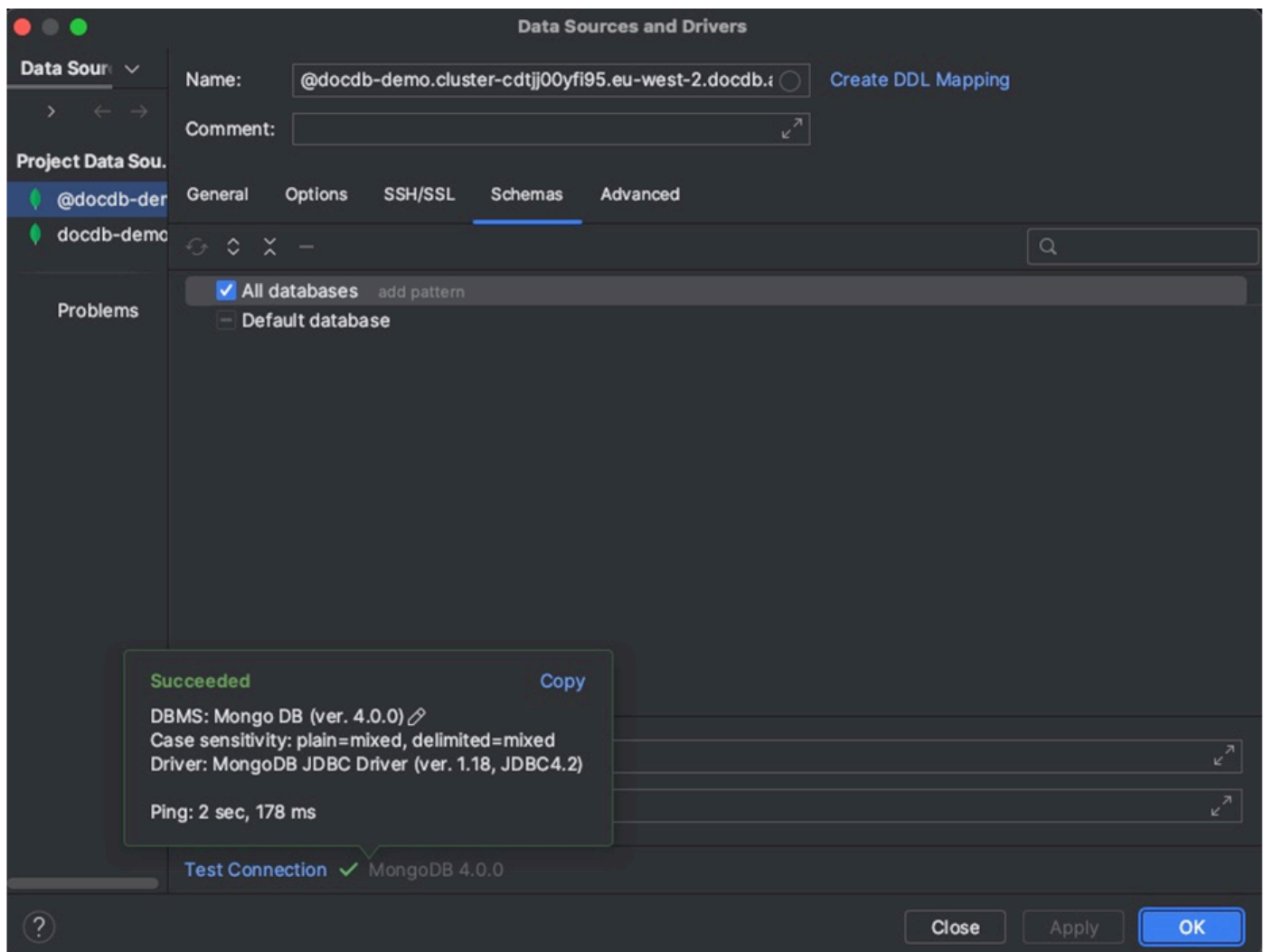
您可以從此位置或使用以下命令下載證書：<https://aws.amazon.com/https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem>

Note

如果您要連線到 Amazon DocumentDB 彈性叢集，則不需要指定 CA 檔案。保持選中「使用 SSL」選項，並將所有其他選項保持默認值。



7. 在 [結構描述] 索引標籤中，選擇 [所有資料庫]，或在 [結構描述模式] 欄位中輸入篩選器「*: *」。按一下測試連線連結以測試連線。



8. 成功測試連線後，按一下確定以儲存資料來源組態。

DataGrip 功能

DataGrip 提供各種功能，協助您有效地使用 Amazon DocumentDB：

- SQL 編輯器 — 使用中的 SQL 編輯器，在 DocumentDB 集合上撰寫和執行類似 SQL 的查詢。
- 視覺化查詢建構器 — 使用視覺化查詢建構器以圖形方式建立查詢，而無需撰寫 SQL 程式。
- 結構描述管理 — 輕鬆管理您的資料庫結構描述，包括建立、變更和刪除集合。
- 資料視覺化 — 使用中提供的各種視覺化工具檢視和分析您的資料 DataGrip。
- 匯出和匯入資料 — 使用 DataGrip 的匯出和匯入功能，在 Amazon DocumentDB 和其他資料庫之間傳輸資料。

如需使用 Amazon DocumentDB 和其他資料庫系統的進階功能和秘訣，請參閱官方 [DataGrip 文件](#)。

使用 Amazon EC2 Connect

本節說明如何設定 Amazon DocumentDB 叢集和 Amazon EC2 之間的連線，以及如何從 Amazon EC2 執行個體存取亞馬遜文件資料庫叢集。

設定 EC2 連線有兩個選項：

- [自動將 EC2 執行個體連線到 Amazon DocumentDB 資料庫](#) — 使用 EC2 主控台內的自動連線功能，自動設定 EC2 執行個體與新的或現有的 Amazon DocumentDB 資料庫之間的連線。此連線允許流量在 EC2 執行個體和 Amazon DocumentDB 資料庫之間傳輸。此選項通常用於測試和建立新的安全群組。
- [手動將 EC2 執行個體連接到 Amazon DocumentDB 資料庫](#) — 透過手動設定和指派安全群組來設定 EC2 執行個體與 Amazon DocumentDB 資料庫之間的連線，以重現自動連線功能建立的組態。此選項通常用於變更更多進階設定，以及使用現有的安全性群組。

必要條件

無論選項為何，在建立第一個 Amazon DocumentDB 叢集之前，都必須執行下列動作：

創建一個 Amazon Web Services (AWS) 帳戶

在您可以開始使用 Amazon DocumentDB 之前，您必須有一個 Amazon Web Services (AWS) 帳戶。該 AWS 帳戶是免費的。您僅需按使用的服務和資源量付費。

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建 AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，[將管理存取權指派給管理使用者](#)，並且僅使用根使用者來執行 [需要根使用者存取權的任務](#)。

視需要設定所需的 AWS Identity and Access Management (IAM) 許可。

存取管理 Amazon DocumentDB 資源 (例如叢集、執行個體和叢集參數群組) 需要 AWS 可用來驗證請求的登入資料。如需詳細資訊，請參閱 [Amazon DocumentDB 的 Identity and Access Management](#)。

1. 在的搜尋列中輸入 IAM AWS Management Console，然後在出現的下拉式功能表中選取 IAM。
2. 進入 IAM 主控台後，從導覽窗格中選取 [使用者]。
3. 選擇您的用戶名。
4. 點擊按鈕添加權限。
5. 選取直接連接現有政策。
6. AmazonDocDBFullAccess在搜索欄中鍵入並在搜索結果中出現時選擇它。
7. 單擊底部顯示「下一步：查看」的藍色按鈕。
8. 點擊底部顯示「添加權限」的藍色按鈕。

創建一個 Amazon Virtual Private Cloud (Amazon VPC)

根據 AWS 區域 您所在的位置，您可能已經建立或可能沒有建立預設 VPC。如果您沒有預設 VPC，請完成 Amazon VPC 使用者指南中的 [Amazon VPC 入門](#) 步驟 1。這將需要不到五分鐘。

自動 Connect Amazon EC2

主題

- [自動將 EC2 執行個體連線到新的亞馬遜資料庫](#)
- [自動將 EC2 執行個體連線到現有的亞馬遜資料庫](#)
- [與 EC2 執行個體自動連線的概要](#)
- [檢視已連線的運算資源](#)

在設定 EC2 執行個體和新 Amazon DocumentDB 資料庫之間的連線之前，請確定您符合中所述的要求。[與 EC2 執行個體自動連線的概要](#) 如果您在設定連線後變更安全群組，這些變更可能會影響 EC2 執行個體和 Amazon DocumentDB 資料庫之間的連線。

Note

您只能在 EC2 執行個體和 Amazon DocumentDB 資料庫之間使用 AWS Management Console。您無法使用 AWS CLI 或 Amazon DocumentDB API 自動設置連接。

自動將 EC2 執行個體連線到新的亞馬遜資料庫

下列程序假設您已完成 [必要條件](#) 主題中的步驟。

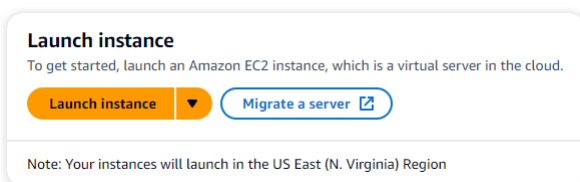
步驟

- [步驟 1：創建一個 Amazon EC2 實例](#)
- [步驟 2：創建一個 Amazon DocumentDB 集群](#)
- [步驟 3：Connect 到您的 Amazon EC2 實例](#)
- [第 4 步：安裝蒙戈外殼](#)
- [第 5 步：管理 Amazon DocumentDB TLS](#)
- [步驟 6：Connect 到您的 Amazon DocumentDB 集群](#)
- [步驟 7：插入和查詢資料](#)
- [步驟 8：探索](#)

步驟 1：創建一個 Amazon EC2 實例

在此步驟中，您將在稍後用來佈建 Amazon DocumentDB 叢集的同一個區域和 Amazon VPC 中建立一個 Amazon EC2 執行個體。

1. 在 Amazon EC2 主控台上，選擇啟動執行個體。



2. 在「名稱和標籤」區段的「名稱」欄位中輸入名稱或識別碼。
3. 在 Amazon 機器映像 (AMI) 下拉列表中，找到 Amazon Linux 2 AMI 並選擇它。

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

Amazon Linux
aws

macOS
Mac

ubuntu

Microsoft

Red Hat

SUSE

debian

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0fa1ca9559f1892ec (64-bit (x86)) / ami-0c80bdc3fa1b47c1f (64-bit (Arm)) Free tier eligible

Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2 Kernel 5.10 AMI 2.0.20231116.0 x86_64 HVM gp2

Architecture **AMI ID**

64-bit (x86) ami-0fa1ca9559f1892ec Verified provider

4. 在「執行個體類型」下拉式清單中找到並選擇 t3.micro。

▼ **Instance type** [Info](#) | [Get advice](#)

Instance type

t3.micro
Family: t3 2 vCPU 1 GiB Memory Current generation: true

On-Demand SUSE base pricing: 0.0104 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour
On-Demand RHEL base pricing: 0.0704 USD per Hour On-Demand Windows base pricing: 0.0196 USD per Hour

All generations [Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

5. 在 [key pair (登入)] 區段中，輸入現有金鑰配對的識別碼，或選擇 [建立新金鑰配對]。

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select [Create new key pair](#)

您必須提供亞 Amazon EC2 key pair。

如果你有一個 Amazon EC2 key pair：

- 選取 key pair，然後從清單中選擇您的 key pair。
- 您必須已經擁有可用的私密金鑰檔案 (.pem 或 .ppk 檔案) 才能登入 Amazon EC2 執行個體。

如果您沒有 Amazon EC2 key pair：

- 選擇建立新 key pair，建立 key pair 話方塊便會顯示。

- b. 在金鑰配對名稱欄位中輸入名稱。
- c. 選擇密鑰對類型和私鑰文件格式。
- d. 選擇 Create key pair (建立金鑰對)。

Create key pair ✕

Key pair name
Key pairs allow you to connect to your instance securely.

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

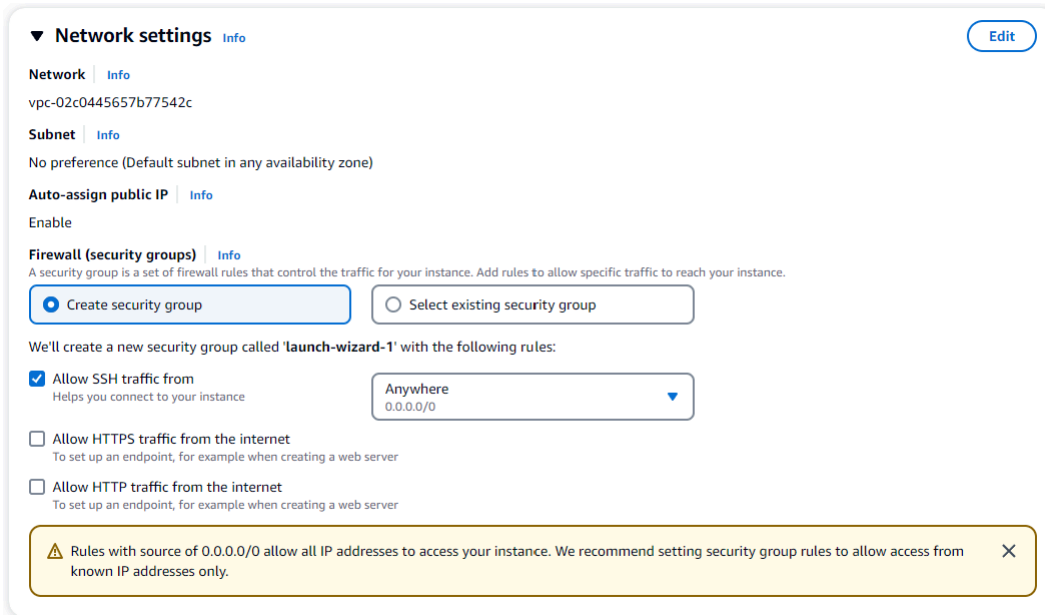
⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#) [↗](#)

[Cancel](#) [Create key pair](#)

i Note

為了安全起見，我們強烈建議您使用金鑰對來進行 SSH 和網際網路連線至 EC2 執行個體。

6. 選用性：在「網路設定」區段的「防火牆 (安全群組)」下，選擇「建立安全性群組」或「選取現有的安全性群組」。



▼ Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-02c0445657b77542c

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from
Helps you connect to your instance Anywhere
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

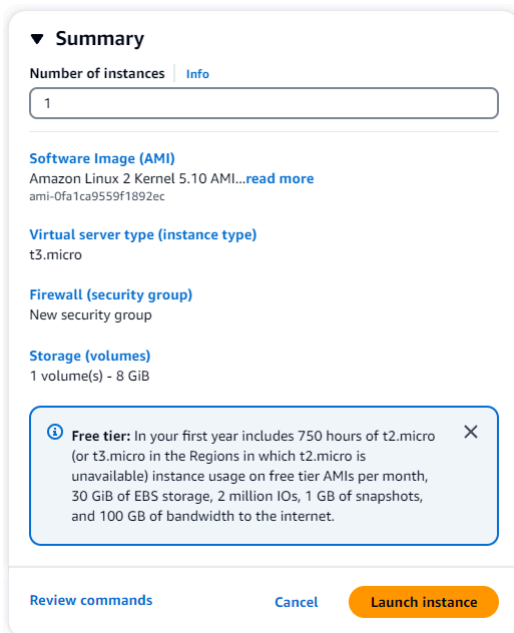
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. [×](#)

如果您選擇選取現有的安全性群組，請從 [一般安全性群組] 下拉式清單中選取一個群組。

如果您選擇建立新的安全群組，請檢查所有適用於 EC2 連線的流量允許規則。

7. 在「摘要」區段中，檢閱您的 EC2 組態，如果正確，請選擇 Launch 執行個體。編輯安全性群組。



▼ Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)
ami-0fa1ca9559f1892ec

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

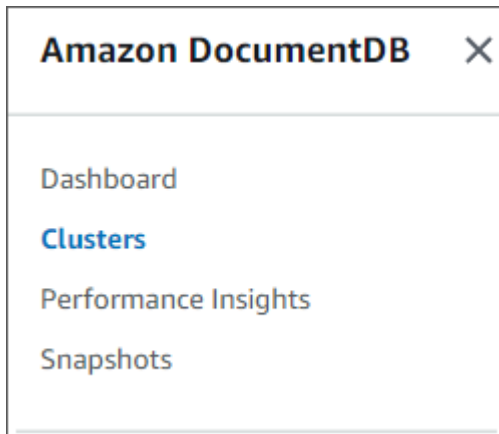
ⓘ **Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet. [×](#)

[Review commands](#) [Cancel](#) [Launch instance](#)

步驟 2：創建一個 Amazon DocumentDB 叢集

在佈建 Amazon EC2 執行個體時，您將建立您的 Amazon DocumentDB 叢集。

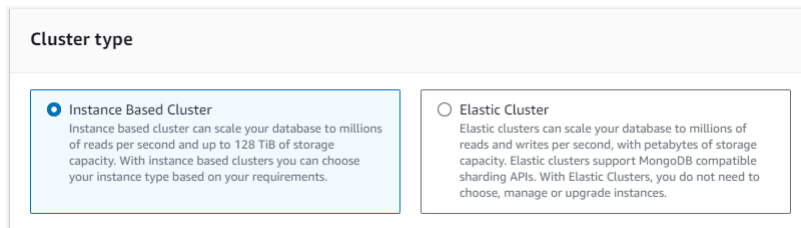
1. 導覽至 Amazon DocumentDB 主控台，然後從導覽窗格中選擇「叢集」。



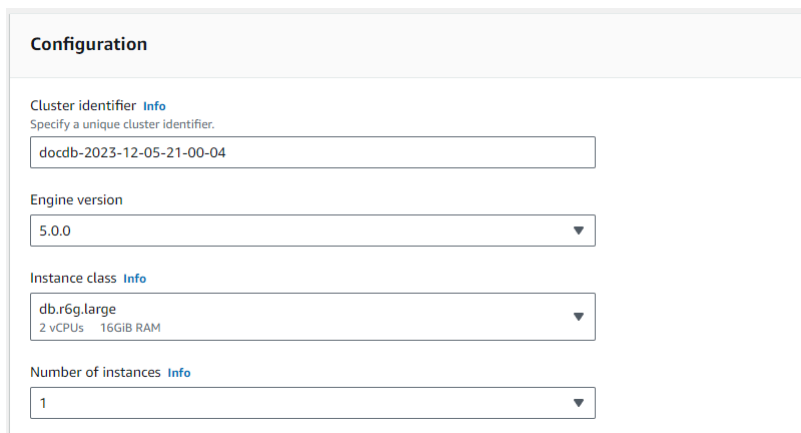
2. 選擇建立。

Create

3. 將 [叢集類型] 設定保留為 [執行個體型叢集] 的預設值。



4. 針對執行個體數目，選擇 1。這將使成本降至最低。將其他設定保留為預設值。



5. 對於連線，請選擇 Connect 線到 EC2 運算資源。這是您在步驟 1 中建立的 EC2 執行個體。

Connectivity ↻

Compute resources
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database.

EC2 Instance
Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i After a database is created, you can't change its VPC.

i Note

連接至 EC2 運算資源會自動為您的 EC2 運算資源連線建立一個安全群組與 Amazon DocumentDB 叢集的連線。完成叢集建立並想要查看新建立的安全性群組後，請瀏覽至叢集清單並選擇叢集的識別碼。在「連線與安全性」標籤中，移至「安全性群組」，然後在「安全性群組名稱 (ID)」下找到您的群組。它將包含您的群組的名稱，並且看起來類似於：docdb-ec2-docdb-2023-12-11-21-33-41:i-0e4bb09985d2bbc4c (sg-0238e0b0bf0f73877)。

6. 對於「驗證」，請輸入登入認證。重要:在稍後的步驟中，您將需要登入認證，才能驗證叢集。

Authentication

Username Info
Specify an alphanumeric string that defines the login ID for the user.

Username must start with a letter and contain 1 to 63 characters

Password Info

Password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).

Confirm password Info

7. 開啟 [顯示進階設定]。

i The estimated hourly cost for 1 db.r6g.large instance(s) is \$0.29/hr. With Amazon DocumentDB you are charged for instances, storage, IOPS, backups, and data transfer. Please see our [pricing page](#) and [cost optimization documentation](#) for more information.

Show advanced settings
 Cancel
Create cluster

8. 在「網路設定」區段中，對於 Amazon VPC 安全群組，請選擇「演示文件資料庫」。

Network settings

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this cluster.

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

Subnet group [Info](#)
A subnet group is a collection of subnets that are within a VPC.

VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

9. 選擇建立叢集。

Create cluster

步驟 3：Connect 到您的 Amazon EC2 實例

要安裝蒙戈外殼，您必須首先連接到您的 Amazon EC2 實例。安裝蒙戈外殼使您能夠連接和查詢您的 Amazon DocumentDB 集群。請完成下列步驟：

1. 在 Amazon EC2 主控台上，導覽至您的執行個體，並查看您剛建立的執行個體是否正在執行。如果是，請按一下執行個體 ID 來選取執行個體。

| <input type="checkbox"/> | Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone |
|-------------------------------------|-----------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|
| <input type="checkbox"/> | aws-cloud9-D... | i-0413cea24ed66b250 | Stopped | t2.micro | - | No alarms | us-east-1c |
| <input checked="" type="checkbox"/> | Sample Server | i-0e4bb09985d2bbc4c | Running | t3.micro | 2/2 checks passed | No alarms | us-east-1a |

2. 選擇連線。

Instance summary for i-0e4bb09985d2bbc4c (Sample Server) Info

Updated less than a minute ago

Refresh
Connect
Instance state ▼
Actions ▼

| | | |
|---|---|--|
| <p>Instance ID i-0e4bb09985d2bbc4c (Sample Server)</p> <p>IPv6 address -</p> <p>Hostname type IP name: ip-172-31-41-131.ec2.internal</p> <p>Answer private resource DNS name IPv4 (A)</p> <p>Auto-assigned IP address 54.87.99.44 [Public IP]</p> <p>IAM Role -</p> <p>IMDSv2 Required</p> | <p>Public IPv4 address 54.87.99.44 [open address]</p> <p>Instance state ● Running</p> <p>Private IP DNS name (IPv4 only) ip-172-31-41-131.ec2.internal</p> <p>Instance type t3.micro</p> <p>VPC ID vpc-02c0445657b77542c</p> <p>Subnet ID subnet-06676048a6487a578</p> | <p>Private IPv4 addresses 172.31.41.131</p> <p>Public IPv4 DNS ec2-54-87-99-44.compute-1.amazonaws.com [open address]</p> <p>Elastic IP addresses -</p> <p>AWS Compute Optimizer finding No recommendations available for this instance.</p> <p>Auto Scaling Group name -</p> |
|---|---|--|

3. 您的連線方法有四個索引標籤式選項：Amazon EC2 執行個體 Connect、工作階段管理員、SSH 用戶端或 EC2 序列主控台。您必須選擇一個並按照其說明進行操作。完成後，選擇「Connect」。

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID
[i-0e4bb09985d2bbc4c](#) (Sample Server)

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
[54.87.99.44](#)

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

i Note

如果您的 IP 位址在您開始本逐步解說後變更，或是稍後再回到您的環境，您必須更新 demoEC2 安全性群組輸入規則，以啟用來自新 API 位址的輸入流量。

第 4 步：安裝蒙戈外殼

您現在可以安裝 mongo 殼層，這是一個命令列公用程式，可用來連接和查詢 Amazon DocumentDB 叢集。請按照以下說明為您的操作系統安裝 mongo 外殼。

On Amazon Linux

在 Amazon Linux 上安裝 mongo 殼層

1. 建立儲存庫檔案。在 EC2 實例的命令行中，執行以下命令：

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://\nrepo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1\nngpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc" | sudo tee /etc/\nyum.repos.d/mongodb-org-5.0.repo
```

2. 當它完成時，通過執行以下命令安裝 mongo 外殼：

```
sudo yum install -y mongodb-org-shell
```

On Ubuntu 18.04

在 Ubuntu 18.04 上安裝 mongo 殼層

1. 匯入公開金鑰供套件管理系統使用。

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv\n2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

2. 以適用於您 Ubuntu 版本的命令來為 MongoDB 建立清單檔案 `/etc/apt/sources.list.d/mongodb-org-3.6.list`。

Ubuntu 18.04

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/\nmongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-\norg-3.6.list
```

Note

上述命令將安裝 Bionic and Xenial 的 mongo 3.6 殼層。

3. 使用下列命令重新載入本機套件資料庫：

```
sudo apt-get update
```

4. 安裝 MongoDB 殼層。

```
sudo apt-get install -y mongodb-org-shell
```

如需在您的 Ubuntu 系統上安裝舊版 MongoDB 的詳細資訊，請參閱[在 Ubuntu 上安裝 MongoDB Community 版本](#)。

On other operating systems

若要在其他作業系統安裝 mongo 殼層，請參閱 MongoDB 文件中的[安裝 MongoDB Community 版本](#)。

第 5 步：管理 Amazon DocumentDB TLS

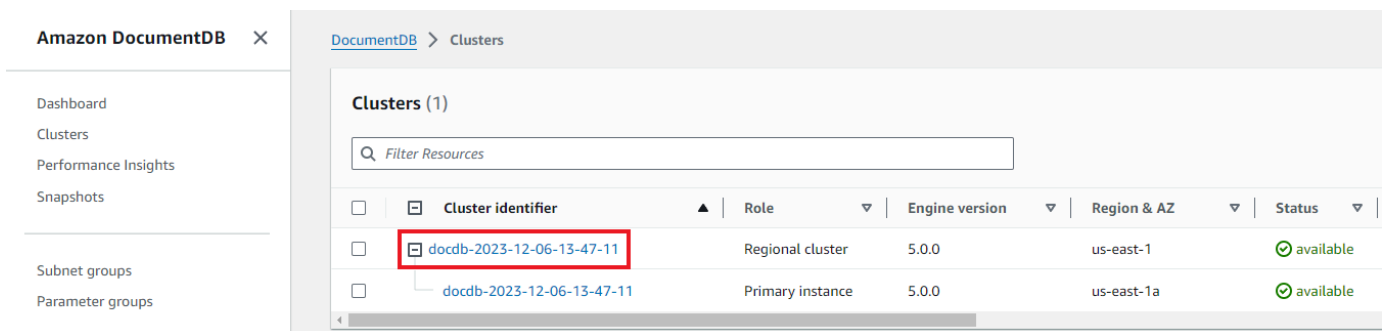
使用以下代碼下載 Amazon DocumentDB 的 CA 證書：`wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

Note

根據預設，任何新的 Amazon DocumentDB 叢集都會啟用傳輸層安全性 (TLS)。如需詳細資訊，請參閱[管理 Amazon DocumentDB 叢集 TLS](#) 設定。

步驟 6：Connect 到您的 Amazon DocumentDB 集群

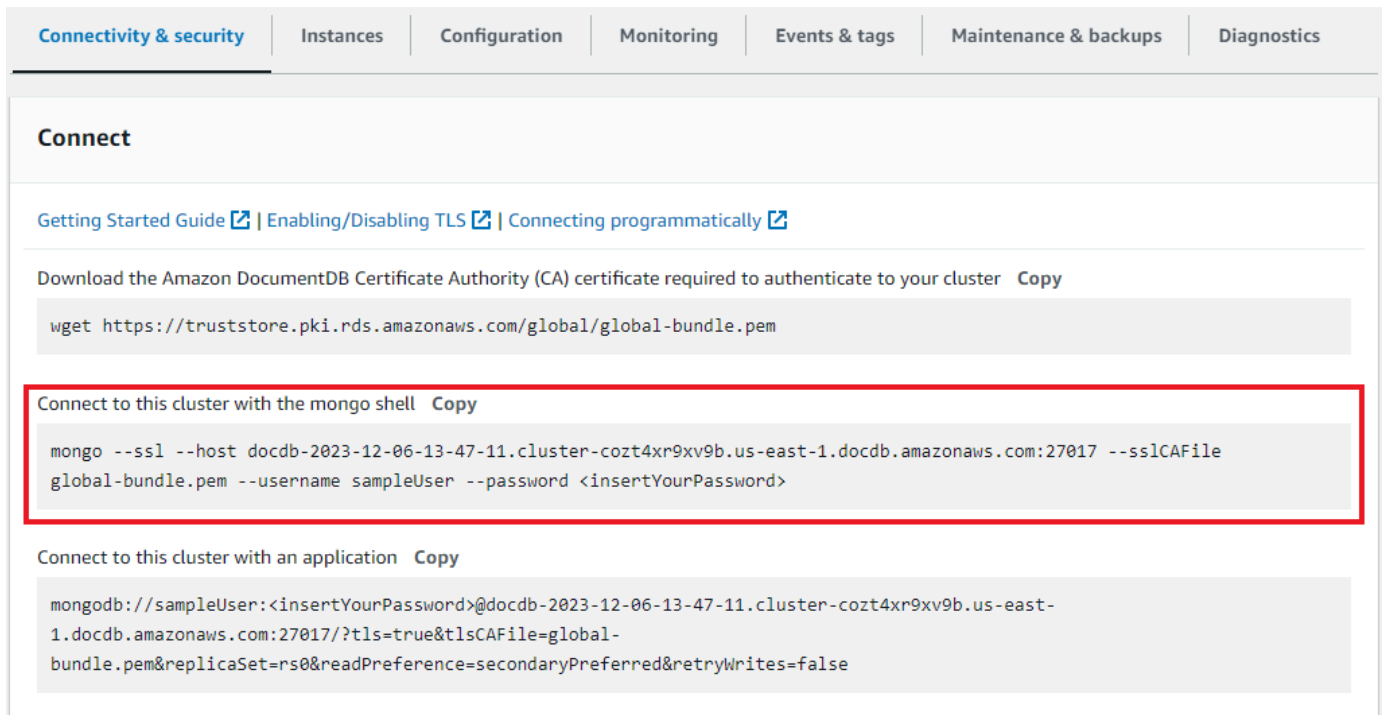
1. 在 Amazon DocumentDB 主控台的叢集下，找到您的叢集。按一下叢集識別碼，選擇您建立的叢集。



The screenshot shows the Amazon DocumentDB console interface. On the left is a navigation menu with options like Dashboard, Clusters, Performance Insights, Snapshots, Subnet groups, and Parameter groups. The main content area is titled 'DocumentDB > Clusters' and shows a table of clusters. The table has columns for Cluster identifier, Role, Engine version, Region & AZ, and Status. One cluster is listed with the identifier 'docdb-2023-12-06-13-47-11', which is highlighted with a red box. This cluster is a 'Regional cluster' with engine version '5.0.0' in the 'us-east-1' region, and its status is 'available'.

| Cluster identifier | Role | Engine version | Region & AZ | Status |
|---------------------------|------------------|----------------|-------------|-----------|
| docdb-2023-12-06-13-47-11 | Regional cluster | 5.0.0 | us-east-1 | available |
| docdb-2023-12-06-13-47-11 | Primary instance | 5.0.0 | us-east-1a | available |

- 在 [連線和安全性] 索引標籤中，找到 [Connect 線] 方塊中使用 mongo 殼層 Connect 線到這個叢集：



The screenshot shows the Amazon DocumentDB console interface. At the top, there are navigation tabs: Connectivity & security, Instances, Configuration, Monitoring, Events & tags, Maintenance & backups, and Diagnostics. The 'Connectivity & security' tab is selected. Below the tabs, the 'Connect' section is visible. It includes links for 'Getting Started Guide', 'Enabling/Disabling TLS', and 'Connecting programmatically'. There are two code blocks: one for downloading the CA certificate using 'wget' and another for connecting to the cluster using the 'mongo' shell. The 'mongo' shell command is highlighted with a red border. Below that, there is a code block for connecting to the cluster using an application.

複製提供的連接字符串並將其粘貼到終端中。

對其進行下列變更：

- 請確定您的字串中有正確的使用者名稱。
- 省略，<insertYourPassword>以便在連接時由 mongo shell 提示您輸入密碼。

您的連接字符串看起來應該類似下列內容：

```
mongo --ssl host docdb-2020-02-08-14-15-11.  
cluster.region.docdb.amazonaws.com:27107 --sslCAFile global-bundle.pem  
--username demoUser --password
```

- 在您的終端中按回車鍵。系統會提示您輸入密碼。輸入您的密碼。
- 當您輸入密碼並看到提示時，表rs0:PRIMARY>示您已成功連線到 Amazon DocumentDB 叢集。

連接時遇到問題？請參閱 [Amazon DocumentDB 疑難排解](#)。

步驟 7：插入和查詢資料

現在您已連線到叢集，您可以執行一些查詢以熟悉文件資料庫的使用。

1. 若要插入單一文件，請輸入下列內容：

```
db.collection.insert({"hello":"DocumentDB"})
```

2. 您會得到以下輸出：

```
WriteResult({ "nInserted" : 1 })
```

3. 您可以讀取使用findOne()指令撰寫的文件 (因為它只會傳回單一文件)。輸入以下內容：

```
db.collection.findOne()
```

4. 您會得到以下輸出：

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" :  
"DocumentDB" }
```

5. 若要執行更多查詢，請考慮使用遊戲設定檔使用案例。首先，將幾個項目插入到標題為的集合中profiles。輸入以下內容：

```
db.profiles.insertMany([  
  { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,  
    "score":202},  
  { "_id" : 2, "name" : "Frank", "status": "inactive", "level": 2,  
    "score":9},  
  { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,  
    "score":87},  
  { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,  
    "score":27}  
])
```

6. 您會得到以下輸出：

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. 使用指find()令傳回縱斷面集合中的所有文件。輸入以下內容：

```
db.profiles.find()
```

8. 您將獲得與您在步驟 5 中輸入的數據匹配的輸出。
9. 使用篩選器查詢單一文件。輸入以下內容：

```
db.profiles.find({name: "Katie"})
```

10. 你應該得到這個輸出：

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
  "score":27}
```

11. 現在，讓我們嘗試查找配置文件並使用findAndModify命令對其進行修改。我們將使用以下代碼給用戶 Matt 額外的十分：

```
db.profiles.findAndModify({
  query: { name: "Matt", status: "active"},
  update: { $inc: { score: 10 } }
})
```

12. 您會得到以下輸出（請注意，他的分數尚未增加）：

```
{
  "_id" : 1,
  "name" : "Matt",
  "status" : "active",
  "level" : 12,
  "score" : 202
}
```

13. 您可以通過以下查詢驗證他的分數是否已更改：

```
db.profiles.find({name: "Matt"})
```

14. 您會得到以下輸出：

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12,
  "score" : 212 }
```

步驟 8：探索

恭喜您！您已成功完成 Amazon DocumentDB 快速入門指南。

下一步是什麼？了解如何充分利用這個功能強大的數據庫以及其一些熱門功能：

- [管理 Amazon DocumentDB](#)
- [擴展](#)
- [備份與還原](#)

Note

若要節省成本，您可以停止 Amazon DocumentDB 叢集以降低成本，也可以刪除叢集。依預設，在閒置 30 分鐘後，您的 AWS Cloud9 環境將停止基礎 Amazon EC2 執行個體。

自動將 EC2 執行個體連線到現有的亞馬遜資料庫

下列程序假設您有一個現有的 Amazon DocumentDB 叢集和一個現有的 Amazon EC2 執行個體。

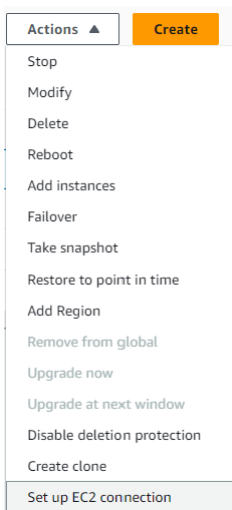
存取您的 Amazon DocumentDB 叢集並設定 Amazon EC2 連線

1. 存取您的 Amazon DocumentDB 叢集。
 - a. [登入 AWS Management Console](#)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
 - b. 在導覽窗格中，選擇叢集。

Tip

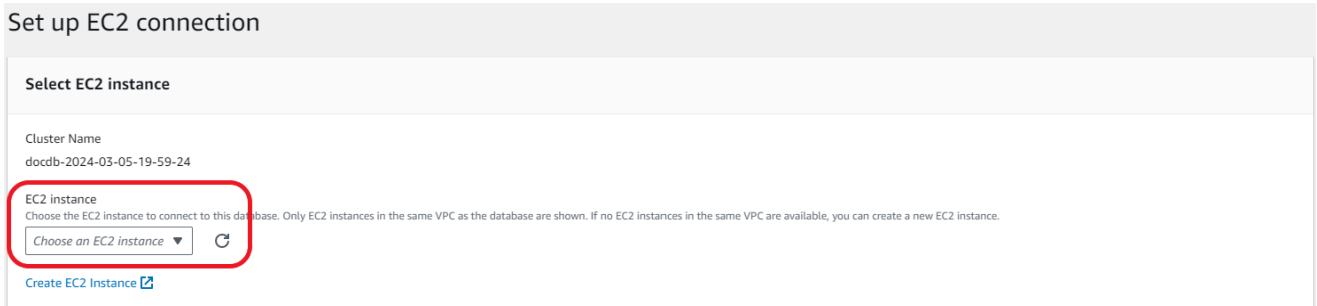
如果畫面左側沒有出現導覽窗格，請選擇頁面左上角的功能表圖示 (☰)。

- c. 選擇叢集名稱左邊的按鈕，指定所需的叢集。
2. 設定 Amazon EC2 連線。
 - a. 選擇 [動作]，然後選擇 [設定 EC2 連線]。



[設定 EC2 連線] 對話方塊隨即出現。

- b. 在 EC2 執行個體欄位中，選擇要連接到叢集的 EC2 執行個體。



- c. 選擇 繼續。

[檢閱並確認] 對話方塊隨即出現

- d. 確保更改正確。然後選擇「設定連線」。

Review and confirm

Connection summary

You are setting up a connection between DocumentDB database docdb-2024-03-05-19-59-24 and EC2 instance i-0413cea24ed66b250

To set up a connection between the database and the EC2 instance, VPC security group docdb-ec2-docdb-2024-03-05-19-59-24-i-0413cea24ed66b250 is added to the DocumentDB cluster, and VPC security group ec2-docdb-docdb-2024-03-05-19-59-24-i-0413cea24ed66b250 is added to the EC2 instance.

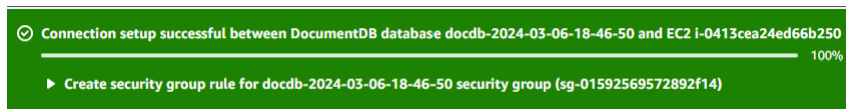
Changes to EC2 instance: i-0413cea24ed66b250

| Attribute | Current value | New value |
|-----------------|--|---|
| Security groups | aws-cloud9-DocumentDBCloud9-9c5f0bc9ff074715afd9d3e4fb7d6fba-InstanceSecurityGroup-1URT6OYVALT77 | aws-cloud9-DocumentDBCloud9-9c5f0bc9ff074715afd9d3e4fb7d6fba-InstanceSecu |

Changes to DocumentDB cluster: docdb-2024-03-05-19-59-24

| Attribute | Current value | New value |
|-----------------|----------------------|---|
| Security groups | sg-021d234a0a3a2c2fe | sg-021d234a0a3a2c2fe, docdb-ec2-docdb-2024-03-05-19-59-24-i-0413cea24ed66b250 |

如果成功，則會出現以下驗證：



與 EC2 執行個體自動連線的概要

當您設定 EC2 執行個體和 Amazon DocumentDB 資料庫之間的連線時，Amazon DocumentDB 會自動為您的 EC2 執行個體和 Amazon DocumentDB 資料庫設定 VPC 安全群組。

以下是將 EC2 執行個體與 Amazon DocumentDB 資料庫連接的要求：

- EC2 執行個體必須存在於與 Amazon 文件資料庫相同的 VPC 中。

如果沒有 EC2 執行個體存在於相同的 VPC 中，則主控台會提供一個連結來建立該執行個體。

- 設定連線的使用者必須擁有執行下列 Amazon EC2 操作的許可：

- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`

- `ec2:RevokeSecurityGroupEgress`

如果資料庫執行個體和 EC2 執行個體位於不同的可用區域，您的帳戶可能會產生跨可用區域成本。

當您設定與 EC2 執行個體的連線時，Amazon DocumentDB 會根據與 Amazon DocumentDB 資料庫和 EC2 執行個體關聯之安全群組的目前組態運作，如下表所述：

| 目前的 Amazon DocumentDB 安全群組組態 | 目前 EC2 安全群組組態 | Amazon DocumentDB 操作 |
|--|---|---|
| <p>有一個或多個與 Amazon DocumentDB 資料庫相關聯的安全群組，其名稱與模式相符。DocumentDB-ec2-n 符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p> | <p>有一個或多個與 EC2 執行個體相關聯的安全群組，其名稱與模式相符 DocumentDB-ec2-n (其中 n 是數字)。符合模式的安全群組尚未修改。此安全群組只有一個輸出規則，將 Amazon DocumentDB 資料庫的 VPC 安全群組做為來源。</p> | <p>Amazon DocumentDB 不採取任何行動。已在 EC2 執行個體和 Amazon DocumentDB 資料庫之間自動設定連線。由於 EC2 執行個體和 Amazon DocumentDB 資料庫之間已存在連線，因此不會修改安全群組。</p> |
| <p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> • 沒有使用符合模式名稱的 Amazon DocumentDB 資料庫相關聯的安全群組。DocumentDB-ec2-n • 有一個或多個與 Amazon DocumentDB 相關聯的安全群組，其名稱與模式相符。DocumentDB-ec2-n 不過，Amazon DocumentDB 無法將這些安全群組中的任何一個用於與 EC2 執行個體的連線。Amazon DocumentDB 無法使用沒有一個輸入規則的安全群組，並將 EC2 執行個體的 VPC 安全群組做為來源。Amazon | <p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> • 沒有與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-DocumtDB-n</code>。 • 有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-DocumtDB-n</code>。不過，亞馬遜文件資料庫無法使用這些安全群組中的任何一個與 Amazon DocumentDB 資料庫的連線。Amazon DocumentDB 無法使用沒有一個輸出規則的安全群組，並將 Amazon DocumentDB 資料庫的 VPC 安全群組做為來源。Amazon DocumentD | <p>Amazon DocumentDB 動作：創建新的安全組</p> |

| 目前的 Amazon DocumentDB 安全群組組態 | 目前 EC2 安全群組組態 | Amazon DocumentDB 操作 |
|---|--|---|
| <p>DocumentDB 也無法使用已修改的安全組。修改範例包括新增規則或變更現有規則的連線埠。</p> | <p>B 也無法使用已修改的安全組。</p> | |
| <p>有一個或多個與 Amazon DocumentDB 資料庫相關聯的安全群組，其名稱與模式相符。DocumentDB-ec2-n 符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p> | <p>有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 ec2-DocumentDB-n。不過，亞馬遜文件資料庫無法使用這些安全群組中的任何一個與 Amazon DocumentDB 資料庫的連線。Amazon DocumentDB 無法使用沒有一個輸出規則的安全群組，並將 Amazon DocumentDB 資料庫的 VPC 安全群組做為來源。Amazon DocumentDB 也無法使用已修改的安全組。</p> | <p>Amazon DocumentDB 動作： 創建新的安全組</p> |
| <p>有一個或多個與 Amazon DocumentDB 資料庫相關聯的安全群組，其名稱與模式相符。DocumentDB-ec2-n 符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p> | <p>存在用於連線的有效 EC2 安全群組，但與 EC2 執行個體沒有相關聯。此安全群組具有符合模式 DocumentDB-ec2-n 的名稱。尚未將其修改。它只有一個輸出規則，將 Amazon DocumentDB 資料庫的 VPC 安全群組做為來源。</p> | <p>Amazon DocumentDB 動作： 關聯 EC2 安全組</p> |

| 目前的 Amazon DocumentDB 安全群組組態 | 目前 EC2 安全群組組態 | Amazon DocumentDB 操作 |
|--|--|--|
| <p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有使用符合模式名稱的 Amazon DocumentDB 資料庫相關聯的安全群組。DocumentDB-ec2-n 有一個或多個與 Amazon DocumentDB 資料庫相關聯的安全群組，其名稱與模式相符。DocumentDB-ec2-n 不過，Amazon DocumentDB 無法將這些安全群組中的任何一個用於與 EC2 執行個體的連線。Amazon DocumentDB 無法使用沒有一個輸入規則的安全群組，並將 EC2 執行個體的 VPC 安全群組做為來源。Amazon DocumentDB 也無法使用已修改的安全組。 | <p>有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 DocumentDB-ec2-n。符合模式的安全群組尚未修改。此安全群組只有一個輸出規則，將 Amazon DocumentDB 資料庫的 VPC 安全群組做為來源。</p> | <p>Amazon DocumentDB 動作： 創建新的安全組</p> |

Amazon DocumentDB 動作：創建新的安全組

Amazon DocumentDB 採取以下操作：

- 建立符合模式 DocumentDB-ec2-n 的新安全群組。此安全群組具有一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。此安全群組與 Amazon DocumentDB 資料庫相關聯，並允許 EC2 執行個體存取亞馬遜文件資料庫。
- 建立符合模式 ec2-DocumentDB-n 的新安全群組。此安全群組具有輸出規則，並將 Amazon DocumentDB 資料庫的 VPC 安全群組做為來源。此安全群組與 EC2 執行個體相關聯，並允許 EC2 執行個體將流量傳送到 Amazon DocumentDB 資料庫。

Amazon DocumentDB 動作：關聯 EC2 安全組

Amazon DocumentDB 將有效的現有 EC2 安全群組與 EC2 執行個體相關聯。此安全群組允許 EC2 執行個體將流量傳送到亞馬遜資料庫。

檢視已連線的運算資源

您可以使用檢視連接 AWS Management Console 到 Amazon DocumentDB 資料庫的運算資源。顯示的資源包括已自動設定的運算資源連線。您可以使用下列方式自動設定與運算資源的連線：

- 您可以在建立資料庫時選取運算資源。如需詳細資訊，請參閱 [創建一個 Amazon DocumentDB 集群](#) 建立異地同步備份資料庫叢集。
- 您可以設定現有資料庫與運算資源之間的連線。如需詳細資訊，請參閱 [自動 Connect Amazon EC2](#)。

列出的運算資源不包括已手動連線至資料庫的運算資源。例如，您可以透過將規則新增至與資料庫關聯的 VPC 安全群組，來允許運算資源手動存取資料庫。

針對要列出的運算資源，必須滿足下列條件：

- 與計算資源關聯的安全群組名稱與模式相符 `ec2-DocumentDB-n` (其中 `n` 是數字)。
- 與運算資源關聯的安全群組具有輸出規則，其連接埠範圍設定為 Amazon DocumentDB 資料庫使用的連接埠。
- 與運算資源關聯的安全群組具有輸出規則，其來源設定為與 Amazon DocumentDB 資料庫關聯的安全群組。
- 與 Amazon DocumentDB 資料庫關聯的安全群組名稱與模式相符 `DocumentDB-ec2-n` (其中 `n` 是數字)。
- 與 Amazon DocumentDB 資料庫關聯的安全群組具有輸入規則，其連接埠範圍設定為 Amazon DocumentDB 資料庫使用的連接埠。
- 與 Amazon DocumentDB 資料庫關聯的安全群組具有輸入規則，其來源設定為與運算資源關聯的安全群組。

若要檢視連線至亞馬遜資料庫的運算資源

1. [登入 AWS Management Console](#)，然後在 <https://console.aws.amazon.com/docdb> 開啟 [Amazon DocumentDB 主控台](#)。
2. 在瀏覽窗格中，選擇 [資料庫]，然後選擇 Amazon DocumentDB 資料庫的名稱。
3. 在 [連線與安全性] 索引標籤上，檢視 [已連線的運算資源] 區段中的運算資源。

手動 Connect Amazon EC2

主題

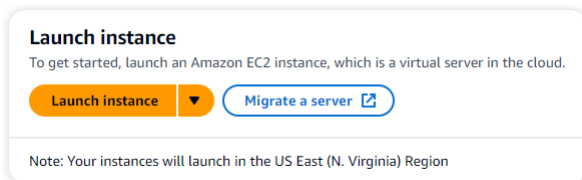
- [步驟 1：創建一個 Amazon EC2 實例](#)
- [步驟 2：建立安全群組](#)
- [第 3 步：創建一個 Amazon DocumentDB 集群](#)
- [步驟 4：Connect 到您的 Amazon EC2 實例](#)
- [第 5 步：安裝蒙戈外殼](#)
- [第 6 步：管理 Amazon DocumentDB TLS](#)
- [步驟 7：Connect 到您的 Amazon DocumentDB 集群](#)
- [步驟 8：插入和查詢資料](#)
- [步驟 9：探索](#)

下列步驟假設您已完成[必要條件](#)主題中的步驟。

步驟 1：創建一個 Amazon EC2 實例

在此步驟中，您將在稍後用來佈建 Amazon DocumentDB 叢集的同一個區域和 Amazon VPC 中建立一個 Amazon EC2 執行個體。

1. 在 Amazon EC2 主控台上，選擇啟動執行個體。



2. 在「名稱和標籤」區段的「名稱」欄位中輸入名稱或識別碼。
3. 在 Amazon 機器映像 (AMI) 下拉列表中，找到 Amazon Linux 2 AMI 並選擇它。

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

Amazon Linux
aws

macOS
Mac

ubuntu

Microsoft

Red Hat

SUSE

debian

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0fa1ca9559f1892ec (64-bit (x86)) / ami-0c80bdc3fa1b47c1f (64-bit (Arm)) Free tier eligible

Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2 Kernel 5.10 AMI 2.0.20231116.0 x86_64 HVM gp2

Architecture **AMI ID**

64-bit (x86) ami-0fa1ca9559f1892ec Verified provider

4. 在「執行個體類型」下拉式清單中找到並選擇 t3.micro。

▼ **Instance type** [Info](#) | [Get advice](#)

Instance type

t3.micro
Family: t3 2 vCPU 1 GiB Memory Current generation: true

On-Demand SUSE base pricing: 0.0104 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour
On-Demand RHEL base pricing: 0.0704 USD per Hour On-Demand Windows base pricing: 0.0196 USD per Hour

All generations [Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

5. 在 [key pair (登入)] 區段中，輸入現有金鑰配對的識別碼，或選擇 [建立新金鑰配對]。

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select [Create new key pair](#)

您必須提供亞 Amazon EC2 key pair。

如果你有一個 Amazon EC2 key pair：

- 選取 key pair，然後從清單中選擇您的 key pair。
- 您必須已經擁有可用的私密金鑰檔案 (.pem 或 .ppk 檔案) 才能登入 Amazon EC2 執行個體。

如果您沒有 Amazon EC2 key pair：

- 選擇建立新 key pair，建立 key pair 話方塊便會顯示。

- b. 在金鑰配對名稱欄位中輸入名稱。
- c. 選擇密鑰對類型和私鑰文件格式。
- d. 選擇 Create key pair (建立金鑰對)。

Create key pair ✕

Key pair name
Key pairs allow you to connect to your instance securely.

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type


RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#) 

[Cancel](#) [Create key pair](#)

i Note

為了安全起見，我們強烈建議您使用金鑰對來進行 SSH 和網際網路連線至 EC2 執行個體。

6. 在 [網路設定] 區段的 [防火牆 (安全性群組)] 下，選擇 [建立安全性群組] 或 [選取現有的安全性群組]。

▼ Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-02c0445657b77542c

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from
Helps you connect to your instance Anywhere
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

如果您選擇選取現有的安全性群組，請從 [一般安全性群組] 下拉式清單中選取一個群組。

如果您選擇建立新的安全性群組，請執行下列動作：

- a. 檢查所有適用於 EC2 連接的流量允許規則。
- b. 在 IP 欄位中，選擇 [我的 IP] 或選取 [自訂] 以從 CIDR 區塊、首碼清單或安全性群組清單中選擇。除非您的 EC2 執行個體位於隔離的網路上，否則我們不建議您選擇 Anywhere，因為它允許任何 IP 地址存取 EC2 執行個體。

My IP
52.95.4.16/32

7. 在「摘要」區段中，檢閱您的 EC2 組態，如果正確，請選擇 Launch 執行個體。編輯安全性群組。

▼ **Summary**

Number of instances [Info](#)

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)
ami-0fa1ca9559f1892ec

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

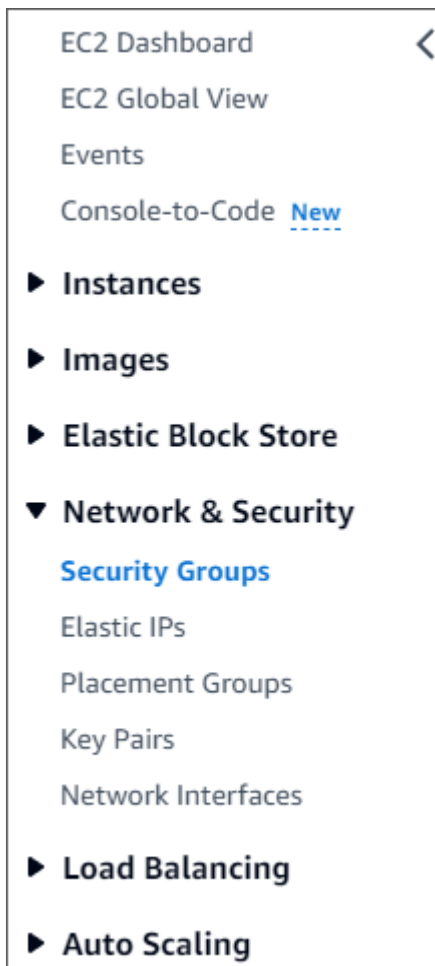
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet. ✕

[Review commands](#) [Cancel](#) [Launch instance](#)

步驟 2：建立安全群組

現在，您將在預設的 Amazon VPC 中建立一個新的安全群組。安全群組demoDocDB可讓您從 Amazon EC2 執行個體連接到連接埠 27017 (亞馬遜文件資料庫的預設連接埠) 上的 Amazon DocumentDB 叢集。

1. 在 [Amazon EC2 管理主控台](#) 的網路和安全性下，選擇安全群組。



2. 選擇建立安全群組。

Create security group

3. 在「基本詳細資料」區段中：
 - a. 針對 Security group name (安全群組名稱)，輸入 demoDocDB。
 - b. 在描述中，輸入描述。
 - c. 對於 VPC，請接受預設 VPC 的使用。

Basic details

Security group name [Info](#)

MyWebServerGroup

Name cannot be edited after creation.

Description [Info](#)

Allows SSH access to developers

VPC Info

vpc-02c0445657b77542c

4. 在 Inbound rules (傳入規則) 區段中，選擇 Add rule (新增規則)。
 - a. 針對 Type (類型)，選擇 Custom TCP Rule (自訂 TCP 規則)。
 - b. 對於連接埠範圍，輸入27017。
 - c. 選擇「自訂」做為「目的地」在旁邊的欄位中，搜尋您剛才呼叫的安全性群組demoEC2。您可能需要為 Amazon EC2 主控台重新整理瀏覽器，才能自動填入demoEC2來源名稱。

Inbound rules [Info](#)

Type [Info](#) Protocol [Info](#) Port range [Info](#) Source [Info](#) Description - optional [Info](#)

Custom TCP TCP 27017 Cust... Q

Add rule Delete

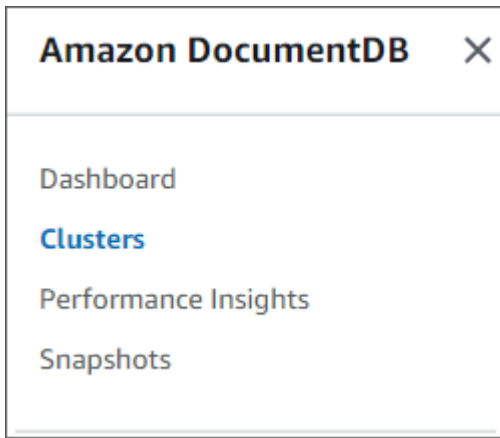
5. 接受所有其他預設值並選擇建立安全性群組。

Create security group

第 3 步：創建一個 Amazon DocumentDB 集群

在佈建 Amazon EC2 執行個體時，您將建立您的 Amazon DocumentDB 叢集。

1. 導覽至 Amazon DocumentDB 主控台，然後從導覽窗格中選擇「叢集」。



2. 選擇建立。

Create

3. 將 [叢集類型] 設定保留為 [執行個體型叢集] 的預設值。

A screenshot of the "Cluster type" section in the Amazon DocumentDB console. It contains two radio button options. The first option, "Instance Based Cluster", is selected and highlighted with a blue border. Its description states: "Instance based cluster can scale your database to millions of reads per second and up to 128 TiB of storage capacity. With instance based clusters you can choose your instance type based on your requirements." The second option, "Elastic Cluster", is not selected. Its description states: "Elastic clusters can scale your database to millions of reads and writes per second, with petabytes of storage capacity. Elastic clusters support MongoDB compatible sharding APIs. With Elastic Clusters, you do not need to choose, manage or upgrade instances."

4. 針對執行個體數目，選擇 1。這將使成本降至最低。將其他設定保留為預設值。

A screenshot of the "Configuration" section in the Amazon DocumentDB console. It includes several input fields and dropdown menus. The "Cluster identifier" field contains the text "docdb-2023-12-05-21-00-04". The "Engine version" dropdown menu is set to "5.0.0". The "Instance class" dropdown menu is set to "db.r6g.large" (with "2 vCPUs 16GiB RAM" listed below it). The "Number of instances" dropdown menu is set to "1".

5. 對於連線，請保留預設設定 [不連線至 EC2 運算資源]。

A screenshot of the "Connectivity" section in the Amazon DocumentDB console. It features a "Compute resources" section with a description: "Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database." There are two radio button options. The first option, "Connect to an EC2 compute resource", is not selected. The second option, "Don't connect to an EC2 compute resource", is selected and highlighted with a blue border. Its description states: "Don't set up a connection to a compute resource for this database."

Note

連線至 EC2 運算資源會自動為您與叢集的 EC2 運算資源連線建立安全群組。由於您在上一個步驟中手動建立這些安全群組，因此您應該選取 [不連線至 EC2 運算資源]，以免建立第二組安全群組。

6. 對於「驗證」，請輸入登入認證。重要:在稍後的步驟中，您將需要登入認證，才能驗證叢集。

Authentication

Username Info
Specify an alphanumeric string that defines the login ID for the user.

Username must start with a letter and contain 1 to 63 characters

Password Info **Confirm password** Info

Password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).

7. 開啟 [顯示進階設定]。

i The estimated hourly cost for 1 db.r6g.large instance(s) is \$0.29/hr. With Amazon DocumentDB you are charged for instances, storage, IOPS, backups, and data transfer. Please see our [pricing page](#) and [cost optimization documentation](#) for more information.

Show advanced settings

8. 在「網路設定」區段中，對於 Amazon VPC 安全群組，請選擇「演示文件資料庫」。

Network settings

Virtual Private Cloud (VPC) Info
VPC defines the virtual networking environment for this cluster.

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

Subnet group Info
A subnet group is a collection of subnets that are within a VPC.

VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

9. 選擇建立叢集。

步驟 4：Connect 到您的 Amazon EC2 實例

要安裝蒙戈外殼，您必須首先連接到您的 Amazon EC2 實例。安裝蒙戈外殼使您能夠連接和查詢您的 Amazon DocumentDB 集群。請完成下列步驟：

1. 在 Amazon EC2 主控台上，導覽至您的執行個體，並查看您剛建立的執行個體是否正在執行。如果是，請按一下執行個體 ID 來選取執行個體。

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone |
|-----------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|
| aws-cloud9-D... | i-0413cea24ed66b250 | Stopped | t2.micro | - | No alarms | us-east-1c |
| Sample Server | i-0e4bb09985d2bbc4c | Running | t3.micro | 2/2 checks passed | No alarms | us-east-1a |

2. 選擇連線。

Instance summary for i-0e4bb09985d2bbc4c (Sample Server)

Updated less than a minute ago

| | | |
|--|---|---|
| Instance ID i-0e4bb09985d2bbc4c (Sample Server) | Public IPv4 address 54.87.99.44 | Private IPv4 addresses 172.31.41.131 |
| IPv6 address - | Instance state Running | Public IPv4 DNS ec2-54-87-99-44.compute-1.amazonaws.com |
| Hostname type IP name: ip-172-31-41-131.ec2.internal | Private IP DNS name (IPv4 only) ip-172-31-41-131.ec2.internal | Elastic IP addresses - |
| Answer private resource DNS name IPv4 (A) | Instance type t3.micro | AWS Compute Optimizer finding No recommendations available for this instance. |
| Auto-assigned IP address 54.87.99.44 [Public IP] | VPC ID vpc-02c0445657b77542c | Auto Scaling Group name - |
| IAM Role - | Subnet ID subnet-06676048a6487a578 | |
| IMDSv2 Required | | |

3. 您的連線方法有四個索引標籤式選項：Amazon EC2 執行個體 Connect、工作階段管理員、SSH 用戶端或 EC2 序列主控台。您必須選擇一個並按照其說明進行操作。完成後，選擇「Connect」。

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-0e4bb09985d2bbc4c (Sample Server)

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
54.87.99.44

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.
ec2-user

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Note

如果您的 IP 位址在您開始本逐步解說後變更，或是稍後再回到您的環境，您必須更新 demo EC2 安全性群組輸入規則，以啟用來自新 API 位址的輸入流量。

第 5 步：安裝蒙戈外殼

您現在可以安裝 mongo 殼層，這是一個命令列公用程式，可用來連接和查詢 Amazon DocumentDB 叢集。請按照以下說明為您的操作系統安裝 mongo 外殼。

On Amazon Linux

在 Amazon Linux 上安裝 mongo 殼層

1. 建立儲存庫檔案。在 EC2 實例的命令行中，執行以下命令：

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://\nrepo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/\nngpgcheck=1 \nenabled=1\nngpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc" | sudo tee /etc/\nyum.repos.d/mongodb-org-5.0.repo
```

2. 當它完成時，通過執行以下命令安裝 mongo 外殼：

```
sudo yum install -y mongodb-org-shell
```

On Ubuntu 18.04

在 Ubuntu 18.04 上安裝 mongo 殼層


1. 匯入公開金鑰供套件管理系統使用。

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv\n2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

2. 以適用於您 Ubuntu 版本的命令來為 MongoDB 建立清單檔案 `/etc/apt/sources.list.d/mongodb-org-3.6.list`。

Ubuntu 18.04

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.6.list
```

 Note

上述命令將安裝 Bionic and Xenial 的 mongo 3.6 殼層。

3. 使用下列命令重新載入本機套件資料庫：

```
sudo apt-get update
```

4. 安裝 MongoDB 殼層。

```
sudo apt-get install -y mongodb-org-shell
```


如需在您的 Ubuntu 系統上安裝舊版 MongoDB 的詳細資訊，請參閱[在 Ubuntu 上安裝 MongoDB Community 版本](#)。

On other operating systems

若要在其他作業系統安裝 mongo 殼層，請參閱 MongoDB 文件中的[安裝 MongoDB Community 版本](#)。

第 6 步：管理 Amazon DocumentDB TLS

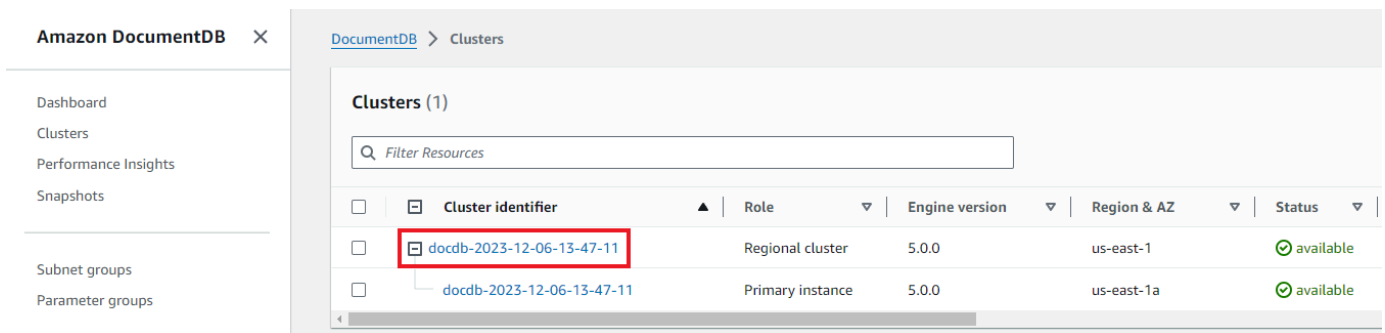
使用以下代碼下載 Amazon DocumentDB 的 CA 證書：`wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

 Note

根據預設，任何新的 Amazon DocumentDB 叢集都會啟用傳輸層安全性 (TLS)。如需詳細資訊，請參閱[管理 Amazon DocumentDB 叢集 TLS 設定](#)。

步驟 7：Connect 到您的 Amazon DocumentDB 集群

1. 在 Amazon DocumentDB 主控台的叢集下，找到您的叢集。按一下叢集識別碼，選擇您建立的叢集。



2. 在 [連線和安全性] 索引標籤中，找到 [Connect 線] 方塊中使用 mongo 殼層 Connect 線到這個叢集：

Connectivity & security | Instances | Configuration | Monitoring | Events & tags | Maintenance & backups | Diagnostics

Connect

[Getting Started Guide](#) | [Enabling/Disabling TLS](#) | [Connecting programmatically](#)

Download the Amazon DocumentDB Certificate Authority (CA) certificate required to authenticate to your cluster [Copy](#)

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Connect to this cluster with the mongo shell [Copy](#)

```
mongo --ssl --host docdb-2023-12-06-13-47-11.cluster-cozt4xr9xv9b.us-east-1.docdb.amazonaws.com:27017 --sslCAFile global-bundle.pem --username sampleUser --password <insertYourPassword>
```

Connect to this cluster with an application [Copy](#)

```
mongodb://sampleUser:<insertYourPassword>@docdb-2023-12-06-13-47-11.cluster-cozt4xr9xv9b.us-east-1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false
```

複製提供的連接字符串並將其粘貼到終端中。

對其進行下列變更：

- a. 請確定您的字串中有正確的使用者名稱。
- b. 省略，<insertYourPassword>以便在連接時由 mongo shell 提示您輸入密碼。

您的連接字串看起來應該類似下列內容：

```
mongo --ssl host docdb-2020-02-08-14-15-11.  
cluster.region.docdb.amazonaws.com:27107 --sslCAFile global-bundle.pem  
--username demoUser --password
```

3. 在您的終端中按回車鍵。系統會提示您輸入密碼。輸入您的密碼。
4. 當您輸入密碼並看到提示時，表rs0:PRIMARY>示您已成功連線到 Amazon DocumentDB 叢集。

連接時遇到問題？請參閱 [Amazon DocumentDB 疑難排解](#)。

步驟 8：插入和查詢資料

現在您已連線到叢集，您可以執行一些查詢以熟悉文件資料庫的使用。

1. 若要插入單一文件，請輸入下列內容：

```
db.collection.insert({"hello":"DocumentDB"})
```

2. 您會得到以下輸出：

```
WriteResult({ "nInserted" : 1 })
```

3. 您可以讀取使用findOne()指令撰寫的文件 (因為它只會傳回單一文件)。輸入以下內容：

```
db.collection.findOne()
```

4. 您會得到以下輸出：

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" :  
"DocumentDB" }
```

5. 若要執行更多查詢，請考慮使用遊戲設定檔使用案例。首先，將幾個項目插入到標題為的集合中profiles。輸入以下內容：

```
db.profiles.insertMany([  
  { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,  
    "score":202},  
  { "_id" : 2, "name" : "Frank", "status": "inactive", "level": 2,  
    "score":9},
```



```
    { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
      "score":87},
    { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
      "score":27}
  ])
```

6. 您會得到以下輸出：

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. 使用指find()令傳回縱斷面集合中的所有文件。輸入以下內容：

```
db.profiles.find()
```

8. 您將獲得與您在步驟 5 中輸入的數據匹配的輸出。

9. 使用篩選器查詢單一文件。輸入以下內容：

```
db.profiles.find({name: "Katie"})
```

10. 你應該得到這個輸出：

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
  "score":27}
```

11. 現在，讓我們嘗試查找配置文件並使用findAndModify命令對其進行修改。我們將使用以下代碼給用戶 Matt 額外的十分：

```
db.profiles.findAndModify({
  query: { name: "Matt", status: "active"},
  update: { $inc: { score: 10 } }
})
```

12. 您會得到以下輸出（請注意，他的分數尚未增加）：

```
{
  "_id" : 1,
  "name" : "Matt",
  "status" : "active",
  "level" : 12,
  "score" : 202
```

```
}
```

13. 您可以通過以下查詢驗證他的分數是否已更改：

```
db.profiles.find({name: "Matt"})
```

14. 您會得到以下輸出：

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12,
  "score" : 212 }
```

步驟 9：探索

恭喜您！您已成功完成 Amazon DocumentDB 快速入門指南。

下一步是什麼？了解如何充分利用這個功能強大的數據庫以及其一些熱門功能：

- [管理 Amazon DocumentDB](#)
- [擴展](#)
- [備份與還原](#)

Note

若要節省成本，您可以停止 Amazon DocumentDB 叢集以降低成本，也可以刪除叢集。依預設，在閒置 30 分鐘後，您的 AWS Cloud9 環境將停止基礎 Amazon EC2 執行個體。

使用 Amazon DocumentDB JDBC 驅動程序進行 Connect

Amazon DocumentDB 的 JDBC 驅動程序為開發人員提供了 SQL 關係界面，並允許從商業智能工具（如 Tableau 和 DbVisualizer）。

如需詳細資訊，請參閱上的 [Amazon DocumentDB JDBC 驅動程式文件 GitHub](#)。

主題

- [入門](#)
- [從桌面 Connect 到 Amazon DocumentDB](#)
- [從 Connect 到 Amazon DocumentDB DbVisualizer](#)

要建立 SSH 隧道，您需要一個 Amazon VPC 中執行的 Amazon EC2 執行個體。您可以使用相同 VPC 中現有的 EC2 執行個體做為您的叢集，或建立一個叢集。您可以在本機電腦上執行下列命令，以設定連至 Amazon DocumentDB 叢集的安全殼層 `sample-cluster.node.us-east-1.docdb.amazonaws.com` 通道。

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -N
```

-L 標誌用於轉發本地端口。這是連線至 VPC 外部用戶端上執行的任何 BI 工具的先決條件。運行上述步驟後，您可以繼續執行所選 BI 工具的後續步驟。

如需 SSH 通道的詳細資訊，請參閱 [使用安全殼層通道連線至 Amazon DocumentDB](#) 的文件。

從桌面 Connect 到 Amazon DocumentDB

主題

- [添加 Amazon DocumentDB JDBC 驅動程序](#)
- [使用畫面連接到 Amazon DocumentDB-SSH 隧道](#)

添加 Amazon DocumentDB JDBC 驅動程序

要從桌面連接到 Amazon DocumentDB，您必須下載並安裝驅動程序和文檔數據庫畫面連接器。

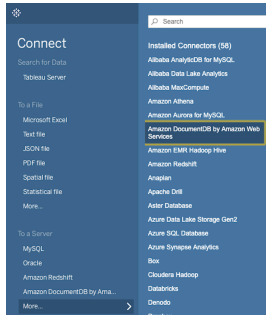
1. 下載 DocumentDB JDBC 驅動程序 JAR 文件，並根據您的操作系統將其複製到以下目錄之一：
 - 視窗-C:\Program Files\Tableau\Drivers
 - MacOS-~/Library/Tableau/Drivers
2. 下載 DocumentDB Tableau 連接器 (TACO 檔案)，並將其複製到「我的 Tableau 儲存庫/連接器」目錄。
 - 視窗-C:\Users\[user]\Documents\My Tableau Repository\Connectors
 - MacOS-/Users/[user]/Documents/My Tableau Repository/Connectors

有關其他資訊，請參閱 [Tableau 文件](#)。

使用畫面連接到 Amazon DocumentDB-SSH 隧道

若要從 DocumentDB 叢集的 VPC 之外的用戶端電腦連線到 Tableau，您必須先設定安全殼層通道，然後再執行以下步驟：

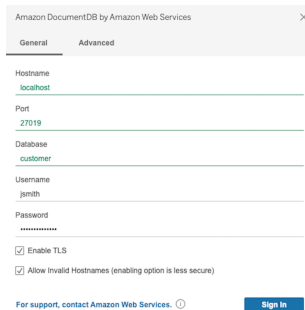
1. 啟動桌面應用程式。
2. 導覽至「Connect」>「至伺服器」>「更多」。
3. 選擇 Amazon DocumentDB 由 Amazon Web Services 下安裝的連接器。



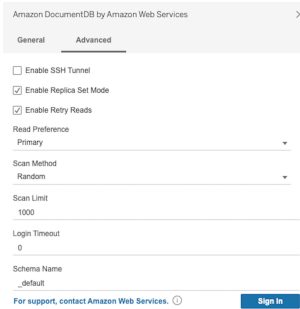
使用畫面連接到 Amazon DocumentDB-外部 SSH 隧道

1. 輸入必要的連線參數主機名稱、連接埠、資料庫、使用者名稱和密碼。以下範例中的連線參數等同於 JDBC 連接字串：

```
jdbc:documentdb://localhost:27019/test?
tls=true&tlsAllowInvalidHostnames=true&scanMethod=random&scanLimit=1000&login
用屬性集中分別傳遞的用戶名和密碼參數。有關連接字符串參數的詳細信息，請參閱 Amazon DocumentDB JDBC 驅動程序 github 文檔。
```



2. (選擇性) 您可以在「進階」索引標籤上找到更多進階選項。



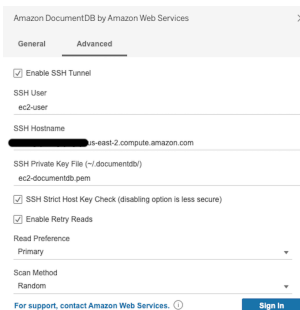
3. 選擇 Sign In (登入)。

使用畫面連接到 Amazon DocumentDB-內部 SSH 隧道

Note

如果您不想使用終端機設定 SSH 通道，可以使用 Tableau GUI 來指定 EC2 執行個體詳細資訊，JDBC 驅動程式本質上將使用這些詳細資料來建立 SSH 通道。

1. 在 [進階] 索引標籤上，選擇 [啟用 SSH 通道] 選項以檢閱其他內容。

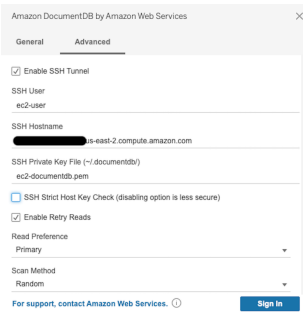


2. 輸入 SSH 使用者、SSH 主機名稱和 SSH 私密金鑰檔案。

3. (選擇性) 您可以停用 SSH 嚴格主機金鑰檢查選項，該選項會略過對已知主機檔案的主機金鑰檢查。

Note

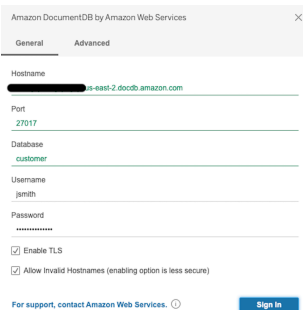
停用此選項的安全性較低，因為它可能會導致 [man-in-the-middle](#) 攻擊。



4. 輸入必要的參數：主機名稱、連接埠、資料庫、使用者名稱和密碼。

Note

使用內部安全殼層通道選項時，請確定您使用 DocumentDB 叢集端點而非本機主機。



5. 選擇 Sign In (登入)。

從 Connect 到 Amazon DocumentDB DbVisualizer

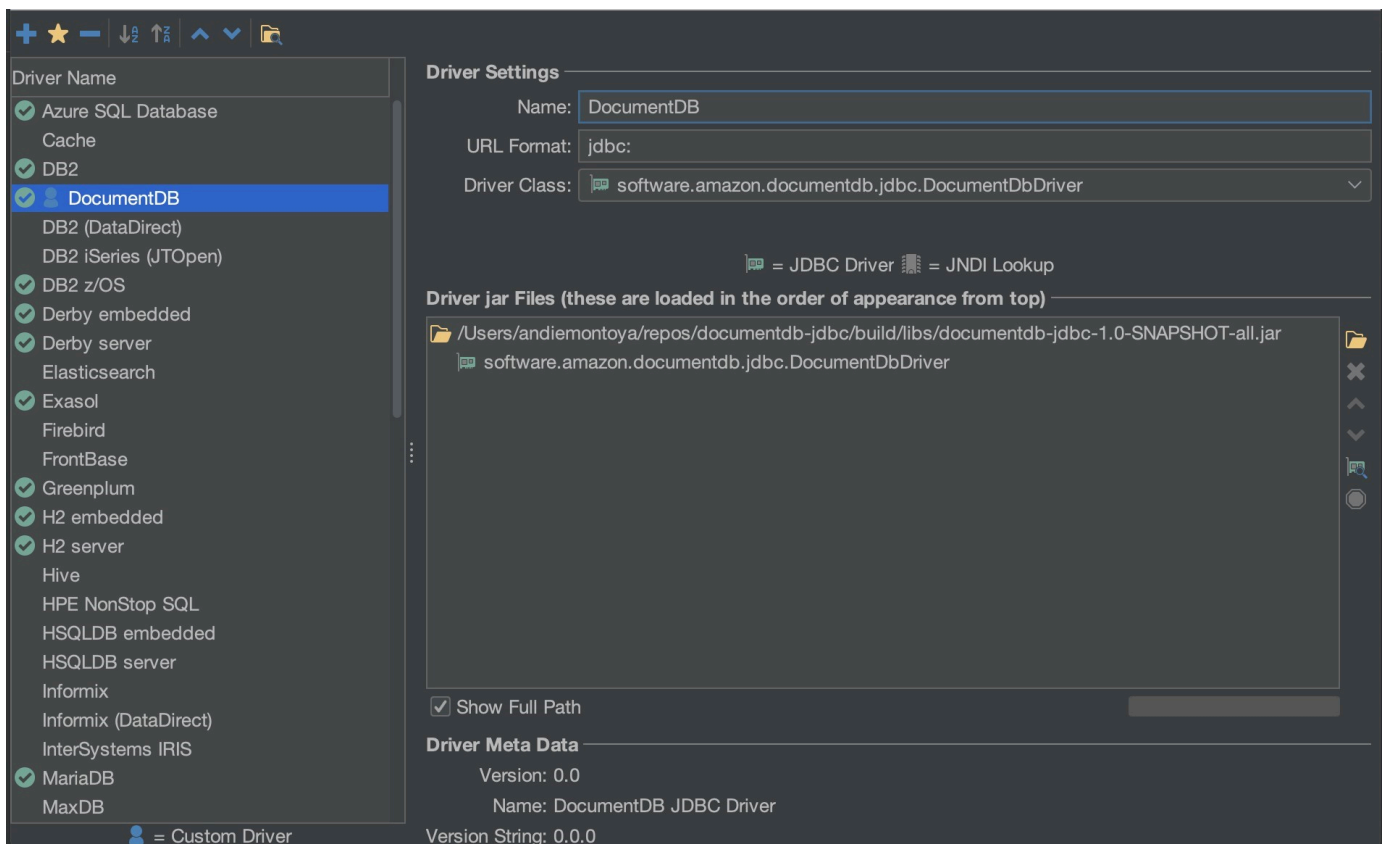
主題

- [添加 Amazon DocumentDB JDBC 驅動程序](#)
- [使用連接到 Amazon DocumentDB DbVisualizer](#)

添加 Amazon DocumentDB JDBC 驅動程序

要連接到 Amazon DocumentDB 從 DbVisualizer 你必須首先導入 Amazon DocumentDB JDBC 驅動程序

1. 啟動 DbVisualizer 應用程式並導航到菜單路徑：工具 > 驅動程序管理器...
2. 選擇 + (或在功能表中選取驅動程式 > 建立驅動程式)。
3. 將 Name (名稱) 設定為 DocumentDB。
4. 將網址格式設定為 `jdbc:documentdb://<host>[:port]/<database>[?option=value[&option=value[...]]]`
5. 選擇資料夾按鈕，然後選取 Amazon DocumentDB JDBC 驅動程式 JAR 檔案，然後選擇開啟按鈕。
6. 確認 [驅動程式類別] 欄位已設定為 `software.amazon.documentdb.jdbc.DocumentDbDriver`。DocumentDB 的驅動程式管理程式設定應如下列範例所示。



7. 關閉對話。Amazon DocumentDB JDBC 驅動程序將被安裝並準備使用。

使用連接到 Amazon DocumentDB DbVisualizer

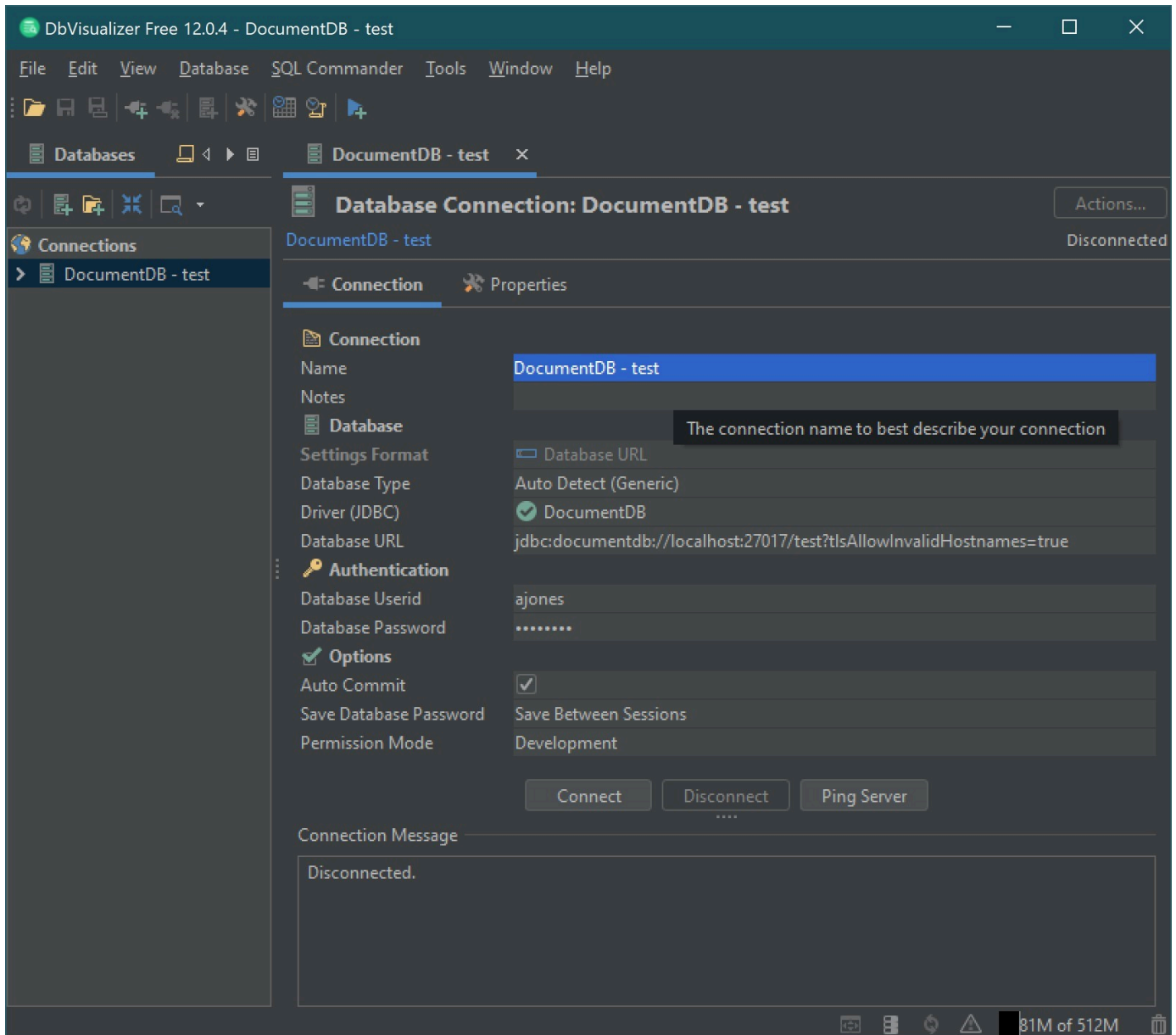
使用 Connect 到 Amazon DocumentDB DbVisualizer

1. 如果您是從 Amazon DocumentDB 叢集的 VPC 人雲端外部進行連線，請確定您已設定安全殼層通道。
2. 從頂層功能表中選擇「資料庫 > 建立資料庫連線」。
3. 在「名稱」欄位中輸入描述性名稱。
4. 將驅動程式 (JDBC) 設定為您在上節中建立的 DocumentDB 驅動程式。
5. 將數據庫 URL 設置為您的 JDBC 連接字符串。

例如：`jdbc:documentdb://localhost:27017/database?
tlsAllowInvalidHostnames=true`

6. 將數據庫用戶標識設置為您的 Amazon DocumentDB 用戶 ID。
7. 將「資料庫密碼」設定為使用者 ID 的對應密碼。

資料庫連線對話方塊應如下列對話方塊所示：



8. 選擇 Connect (連線)。

JDBC 自動產生結構描述

Amazon DocumentDB 是一個文檔數據庫，因此沒有表和模式的概念。但是，諸如 Tableau 之類的 BI 工具將期望它連接的數據庫以呈現結構描述。具體來說，當 JDBC 驅動程序連接需要獲取數據庫中集合的模式時，它將輪詢數據庫中的所有集合。驅動程序將判斷該集合的結構描述的快取版本是否已經存在。如果快取版本不存在，它會針對文件集合取樣，並根據下列行為建立結構描述。

主題

- [架構產生限制](#)
- [掃描方法選項](#)
- [Amazon DocumentDB](#)
- [對應純量文件欄位](#)
- [物件和陣列資料類型處理](#)

架構產生限制

該 DocumentDB JDBC 驅動程序在 128 個字符的標識符的長度施加限制。模式生成器可以截斷生成的標識符 (表名和列名) 的長度，以確保它們符合該限制。

掃描方法選項

取樣行為可以使用連接字串或資料來源選項進行修改。

- 掃描方法 = <option>
 - random-(預設值)-樣本文件會以隨機順序傳回。
 - IDForward-樣本文檔按 ID 的順序返回。
 - iDreVerse-樣本文件以 id 的相反順序返回。
 - all-抽樣集中的所有文檔。
- 掃描限制 = <n>-要取樣的文件數量。其值必須為正整數。預設值為 1000. 如果「掃描方法」設定為「全部」，則會忽略此選項。

Amazon DocumentDB

該 DocumentDB 服務器支持一些 MongoDB 的數據類型。下面列出的是支持的數據類型，及其相關的 JDBC 數據類型。

| MongoB 資料庫資料庫 | DocumentDB 中支援 | JDBC 資料類型 |
|---------------|----------------|-----------|
| 二進位資料 | 是 | VARBINARY |
| Boolean | 是 | BOOLEAN |
| Double | 是 | DOUBLE |

| MongoB 資料庫資料庫 | DocumentDB 中支援 | JDBC 資料類型 |
|------------------|----------------|-----------|
| 32 位元整數 | 是 | INTEGER |
| 64 位整數 | 是 | BIGINT |
| 字串 | 是 | VARCHAR |
| ObjectId | 是 | VARCHAR |
| 日期 | 是 | TIMESTAMP |
| Null | 是 | VARCHAR |
| 常規表達式 | 是 | VARCHAR |
| 時間戳記 | 是 | VARCHAR |
| MinKey | 是 | VARCHAR |
| MaxKey | 是 | VARCHAR |
| 物件 | 是 | 虛擬表 |
| Array (陣列) | 是 | 虛擬表 |
| Decimal128 | 否 | DECIMAL |
| JavaScript | 否 | VARCHAR |
| JavaScript (有範圍) | 否 | VARCHAR |
| 未定義 | 否 | VARCHAR |
| 符號 | 否 | VARCHAR |
| 數據庫指針 (4.0 +) | 否 | VARCHAR |

對應純量文件欄位

當掃描集中的文檔樣本時，JDBC 驅動程序將創建一個或多個模式來表示集中的樣本。一般而言，文件中的純量欄位會對應至資料表結構定義中的資料行。例如，在一個名為 team 的集合和單個文檔中 { "_id" : "112233", "name" : "Alastair", "age": 25 }，這將映射到模式：

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|-------|-------|---------|-----|
| 球隊 | 團隊識別碼 | VARCHAR | PK |
| 球隊 | name | VARCHAR | |
| 球隊 | 年齡 | INTEGER | |

資料類型衝突推進

掃描取樣文件時，某個欄位的資料類型可能在文件間不一致。在這種情況下，JDBC 驅動程序將 JDBC 數據類型提升為一個共同的數據類型，這將適合從採樣文檔中的所有數據類型。

例如：

```
{
  "_id" : "112233",
  "name" : "Alastair", "age" : 25
}

{
  "_id" : "112244",
  "name" : "Benjamin",
  "age" : "32"
}
```

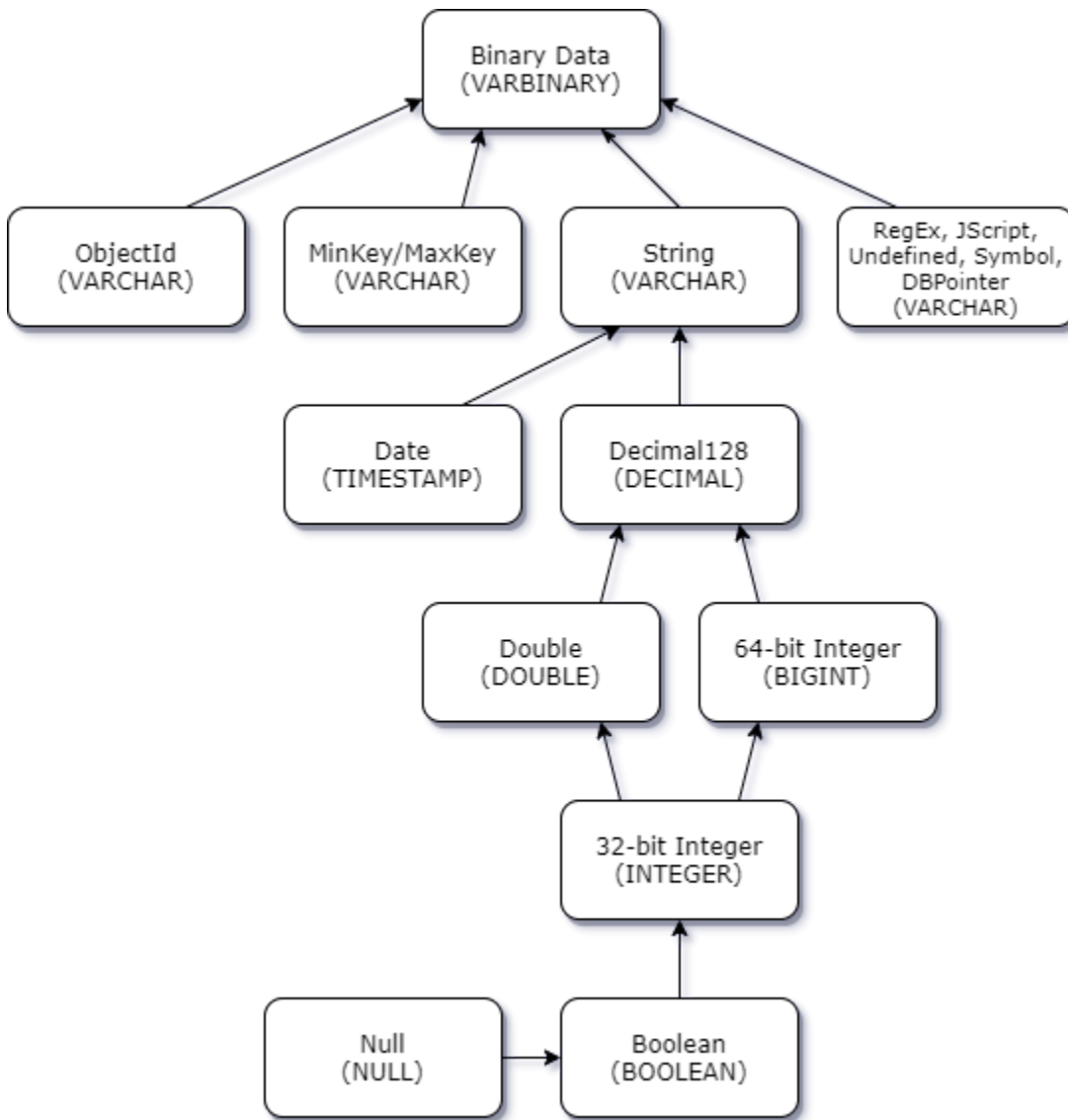
age 字段在第一個文檔中的類型為 32 位整數，但在第二個文檔中的字符串。在這裡，JDBC 驅動程序將 JDBC 數據類型提升為 VARCHAR 遇到時處理任一數據類型。

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|-------|-------|---------|-----|
| 球隊 | 團隊識別碼 | VARCHAR | PK |
| 球隊 | name | VARCHAR | |

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|-------|-------|---------|-----|
| 球隊 | 年齡 | VARCHAR | |

純量衝突推廣

下圖顯示了解決純量純量數據類型衝突的方式。



純量複雜型衝突推廣

就像純量標量類型衝突一樣，不同文檔中的相同字段可能在複雜（數組和對象）和標量（整數，布爾等）之間存在衝突的數據類型。所有這些衝突都會針對這些欄位解決（升級）為 VARCHAR。在此情況下，陣列和物件資料會以 JSON 表示形式傳回。

嵌入式數組-字符串字段衝突示例：

```
{
  "_id": "112233",
  "name": "George Jackson",
  "subscriptions": [
    "Vogue",
    "People",
    "USA Today"
  ]
}
{
  "_id": "112244",
  "name": "Joan Starr",
  "subscriptions": 1
}
```

上面的例子映射到模式的客戶 2 表：

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|-------|----------|---------|-----|
| 客戶 2 | 客戶 2 識別碼 | VARCHAR | PK |
| 客戶 2 | name | VARCHAR | |
| 客戶 2 | 訂閱 | VARCHAR | |

和客戶 1_ 訂閱虛擬表：

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|------------------|------------|---------|-------|
| 客戶 1_ 訂閱 | 客戶 1 識別碼 | VARCHAR | PK/FK |
| 客戶 1_ 訂閱 | 訂閱_索引_LVLO | BIGINT | PK |
| 客戶 1_ 訂閱 | value | VARCHAR | |
| customer_address | city | VARCHAR | |
| customer_address | region | VARCHAR | |

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|------------------|---------|---------|-----|
| customer_address | country | VARCHAR | |
| customer_address | code | VARCHAR | |

物件和陣列資料類型處理

到目前為止，我們只描述了標量數據類型的映射方式。對象和數組數據類型（當前）映射到虛擬表。JDBC 驅動程序將創建一個虛擬表來表示文檔中的對象或數組字段。映射虛擬表的名稱將連接原始集合的名稱，後跟字段的名稱以下劃線字符（「_」）分隔。

基底資料表的主索引鍵（「_id」）會在新虛擬資料表中使用新名稱，並以外部索引鍵的形式提供給關聯的基底資料表。

對於內嵌陣列類型欄位，會產生索引資料行以表示陣列中每個層級的索引。

嵌入式物件欄位範例

對於文件中的物件欄位，JDBC 驅動程式會建立至虛擬資料表的對應。

```
{
  "Collection: customer",
  "_id": "112233",
  "name": "George Jackson",
  "address": {
    "address1": "123 Avenue Way",
    "address2": "Apt. 5",
    "city": "Hollywood",
    "region": "California",
    "country": "USA",
    "code": "90210"
  }
}
```

上面的例子映射到模式客戶表：

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|----------|-------|---------|-----|
| customer | 客戶識別碼 | VARCHAR | PK |

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|----------|-------|---------|-----|
| customer | name | VARCHAR | |

和客戶地址虛擬表：

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|------------------|---------|---------|-------|
| customer_address | 客戶識別碼 | VARCHAR | PK/FK |
| customer_address | 地址 1 | VARCHAR | |
| customer_address | 地址 2 | VARCHAR | |
| customer_address | city | VARCHAR | |
| customer_address | region | VARCHAR | |
| customer_address | country | VARCHAR | |
| customer_address | code | VARCHAR | |

嵌入式陣列欄位範例

對於文件中的陣列欄位，JDBC 驅動程式也會建立至虛擬資料表的對應。

```
{
  "Collection: customer1",
  "_id": "112233",
  "name": "George Jackson",
  "subscriptions": [
    "Vogue",
    "People",
    "USA Today"
  ]
}
```

上面的例子映射到模式客戶 1 表：

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|-------|----------|---------|-----|
| 客戶 1 | 客戶 1 識別碼 | VARCHAR | PK |
| 客戶 1 | name | VARCHAR | |

和客戶 1_訂閱虛擬表：

| 資料表名稱 | 資料行名稱 | 資料類型 | 索引鍵 |
|------------------|------------|---------|-------|
| 客戶 1_訂閱 | 客戶 1 識別碼 | VARCHAR | PK/FK |
| 客戶 1_訂閱 | 訂閱_索引_LVL0 | BIGINT | PK |
| 客戶 1_訂閱 | value | VARCHAR | |
| customer_address | city | VARCHAR | |
| customer_address | region | VARCHAR | |
| customer_address | country | VARCHAR | |
| customer_address | code | VARCHAR | |

SQL 支援和限制性

Amazon DocumentDB JDBC 驅動程序是一個只讀驅動程序，支持 SQL-92 的子集和一些常見的擴展。如需詳細資訊，請參閱 [SQL 限制文件](#) 和 [JDBC 限制文件](#)。

疑難排解

如果您在使用 Amazon DocumentDB JDBC 驅動程式時遇到問題，請參閱 [疑難排解指南](#)。

使用 Amazon DocumentDB ODBC 驅動程序進行 Connect

對於 Amazon DocumentDB ODBC 驅動程序提供了一個 SQL 關係界面的開發人員，並允許從 BI 工具，如電力 BI 桌面和微軟 Excel 的連接。

如需詳細資訊，請參閱上的 [Amazon DocumentDB ODBC 驅動程式文件 GitHub](#)。

主題

- [入門](#)
- [在視窗中設置 Amazon DocumentDB ODBC 驅動程序](#)
- [從微軟 Excel Connect 到 Amazon DocumentDB](#)
- [從微軟電力 BI 桌面 Connect 到 Amazon DocumentDB](#)
- [自動生成模式](#)
- [SQL 支援和限制](#)
- [疑難排解](#)

入門

步驟 1. 創建 Amazon DocumentDB 集群

如果還沒有 Amazon DocumentDB 叢集，有許多方法可以開始使用。

Note

Amazon DocumentDB Virtual Private Cloud (VPC) ate vate ate ate ate vate 如果從叢集 VPC 外部的本機機器連線，則需要建立與 Amazon EC2 執行個體的 SSH 連線。在此情況下，請使用與 [EC2 Connect](#) 中的指示啟動叢集。請參閱[使用安全殼層通道 Connect 到 Amazon DocumentDB](#)，以取得有關 SSH 通道以及何時需要的詳細資訊。

步驟 2. JRE 或 JDK 安裝

視您的 BI 應用程式而定，您可能需要確保在電腦上安裝 64 位元 JRE 或 JDK 安裝版本 8 或更新版本。您可以[在此處](#)下載 Java SE 運行時環境 8。

步驟三. 下載 Amazon DocumentDB ODBC 驅動程序

[在此處](#)下載 Amazon DocumentDB ODBC 驅動程序。選擇正確的安裝程式 (例如，文件-odbc-1.0.0.msi)。請遵循安裝指南。

步驟 4. 使用 SSH 隧道 Connect 到 Amazon DocumentDB

Amazon Virtual Private vate vate vate vate vate vate ate vate ate vate ate vate ate VPC ate ate vate 它們可以由 Amazon EC2 執行個體或部署在相同 Amazon VPC 中的其他AWS服務直接存

取。此外，Amazon EC2 執行個體或相同區域或其他AWS區域中不同 VPC 中的其他AWS服務可透過 VPC 對等存取 Amazon DocumentDB。

但是，假設您的使用案例需要您 (或您的應用程式) 從叢集的 VPC 外部存取 Amazon DocumentDB 資源。對於大多數使用者而言，並非在與 Amazon DocumentDB 叢集相同 VPC 中的虛擬機器上執行其應用程式，就會發生這種情況。從 VPC 外部連線時，您可以使用 SSH 通道 (也稱為連接埠轉送) 來存取您的 Amazon DocumentDB 資源。

若要建立 SSH 執行個體，您需要 Amazon EC2 執行個體與 Amazon Virtual Private 相同的 Amazon VPC 中執行。您可以使用相同 VPC 中現有的 EC2 執行個體做為您的叢集，或建立一個叢集。您可以在本機電腦上執行下列命令，以設定連至 Amazon DocumentDB 叢集的安全殼層 `sample-cluster.node.us-east-1.docdb.amazonaws.com` 通道：

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -N
```

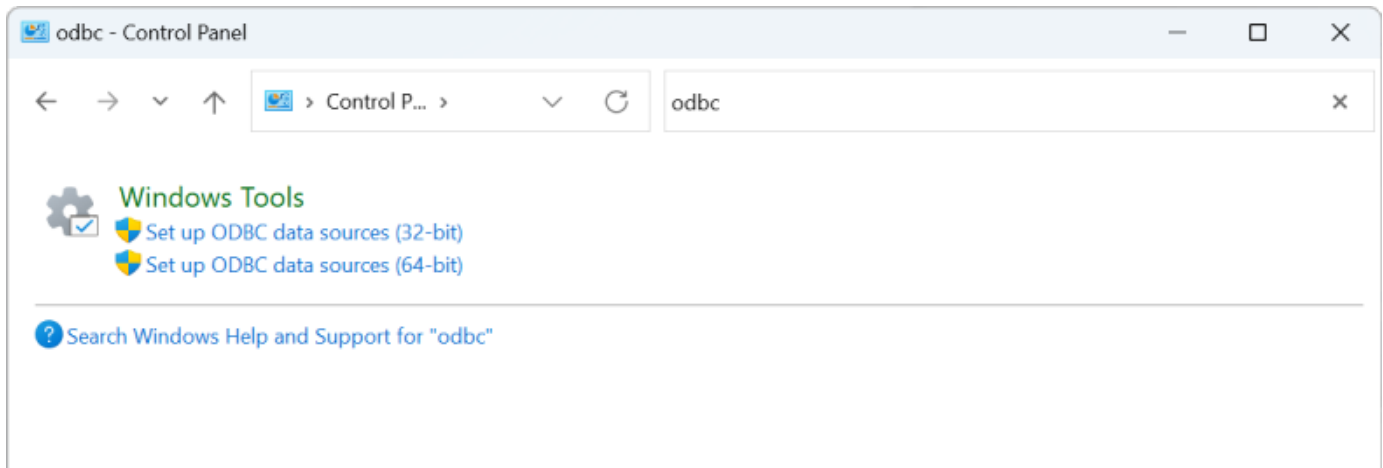
-L 旗標用於轉送本機連接埠。這是連線至 VPC 外部用戶端上執行的任何 BI 工具的先決條件。運行上述步驟後，您可以繼續執行所選 BI 工具的後續步驟。

如需有關安全殼層通道的進一步資訊，請參閱[使用安全殼層通道 Connect 至 Amazon DocumentDB](#) 的文件。

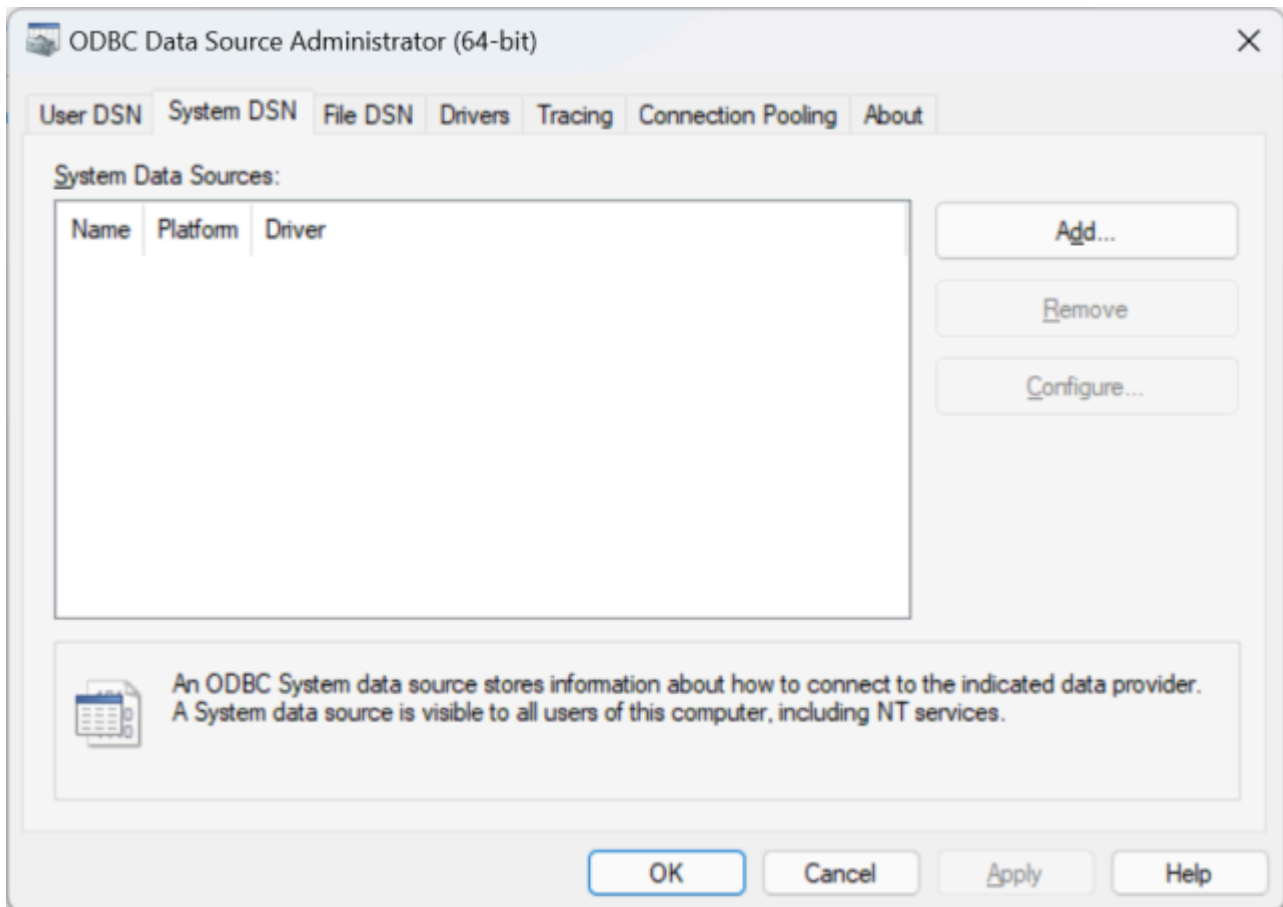
在視窗中設置 Amazon DocumentDB ODBC 驅動程序

使用下列程序以在視窗中設定 Amazon DocumentDB 驅動程式：

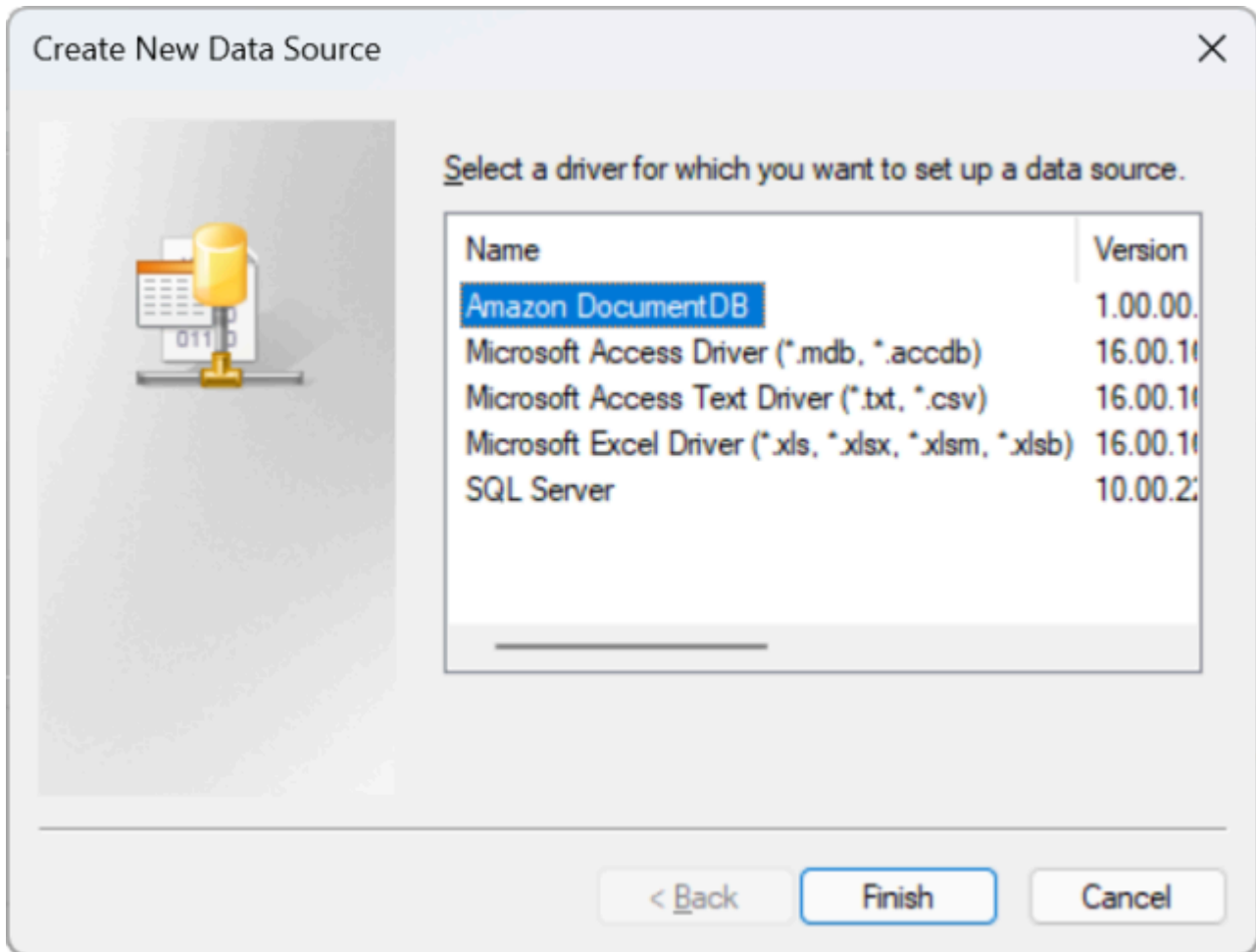
1. 在視窗中開啟控制台並搜尋 ODBC (或在功能表中選取視窗工具 > ODBC 資料來源 (32 位元) 或 ODBC 資料來源 (64 位元))：



2. 選取適當的 ODBC 驅動程式資料來源管理員：如果已安裝，請選擇 32 位元版本，否則請選擇 64 位元版本。
3. 選取系統 DSN 索引標籤，然後按一下新增... 新增 DSN：



4. 從資料來源驅動程式清單中選擇 Amazon 文件資料庫：



5. 在設定 Amazon DocumentDB DSN 對話方塊中，完成組態設定、TLS 索引標籤和測試連線欄位，然後按一下儲存：

Configure Amazon DocumentDB DSN

Connection Settings

Data Source Name*: DocumentDB DSN

Hostname*: docdb-2023-04-09-00-13-17.cpluojuahk1k.us-east-2.docdb.amazonaws.c

Port*: 27017

Database*: employees

TLS SSH Tunnel Schema Logging Additional

Enable TLS

Allow Invalid Hostnames (enabling option is less secure)

TLS CA File: C:\Users\narek\global-bundle.pem

Test Connection

User: adminadmin

Password: ●●●●●●●●●●

Enter valid User and Password to test the connection settings. Test

Version: 1.0.0

Save Cancel

- 請務必準確地完成 Windows 表單，因為連線詳細資料會因您選擇到 EC2 執行個體的 SSH 通道方法而有所不同。請參閱[這裡](#)的 SSH 隧道方法。如需有關每個屬性的詳細資訊，請參閱[連接字串語法和選項](#)。

Configure Amazon DocumentDB DSN

Connection Settings

Data Source Name*: DocumentDB DSN

Hostname*: docdb-2023-04-09-00-13-17.cpluojuahk1k.us-east-2.docdb.amazonaws.c

Port*: 27017

Database*: employees

TLS SSH Tunnel Schema Logging Additional

Enable SSH Tunnel

SSH User: ec2-user

SSH Hostname: ec2-18-221-174-48.us-east-2.compute.amazonaws.com

SSH Private Key File: C:\Users\narek\docdbec2keypair.pem ...

SSH Strict Host Key Check (disabling option is less secure)

SSH Known Hosts File: ...

Test Connection

User: adminadmin

Password:

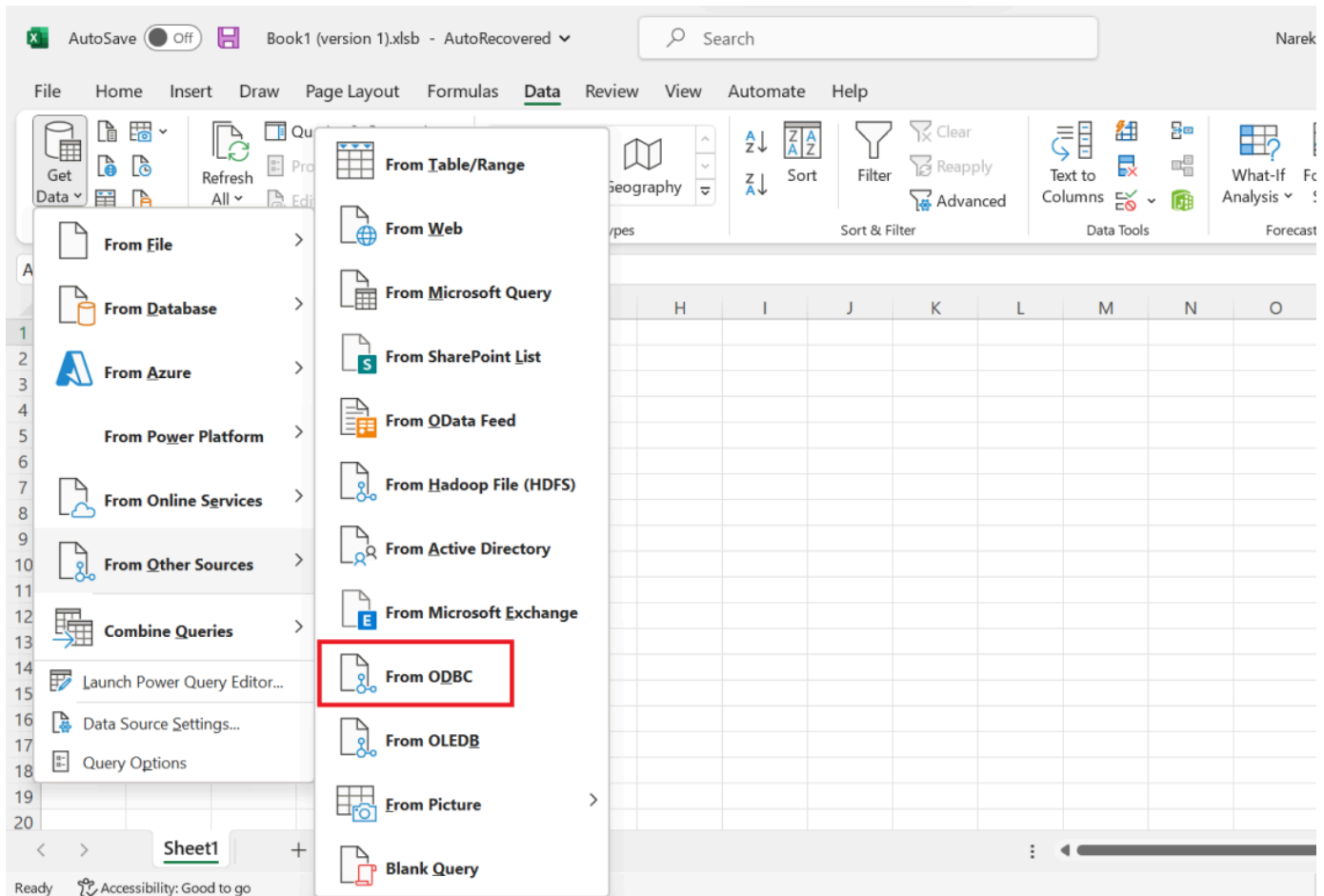
Enter valid User and Password to test the connection settings. Test

Version: 1.0.0 Save Cancel

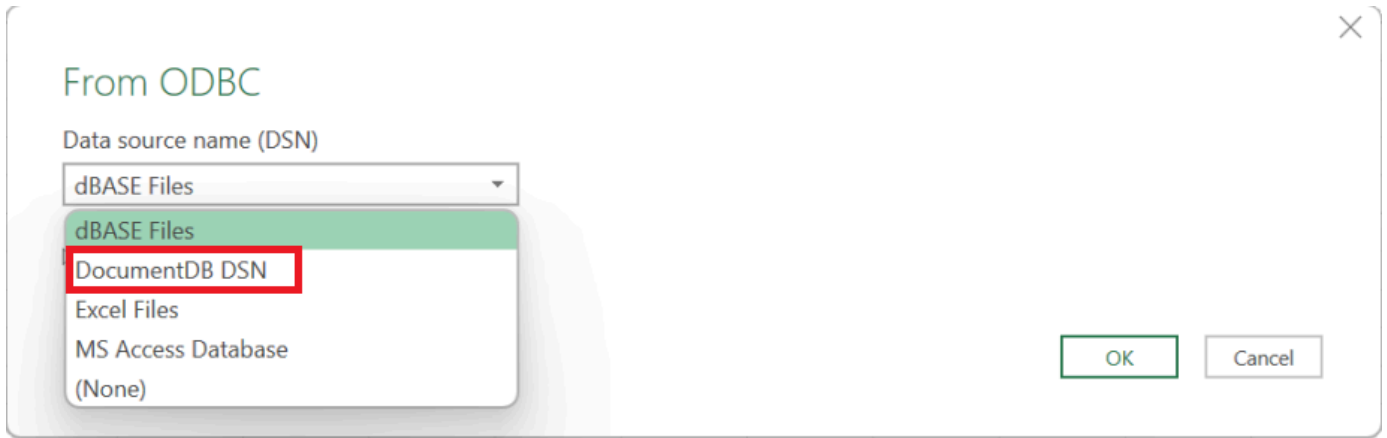
如需有關在 Windows 上設定 Amazon DocumentDB ODBC 驅動程式的詳細資訊，請按一下[這裡](#)。

從微軟 Excel Connect 到 Amazon DocumentDB

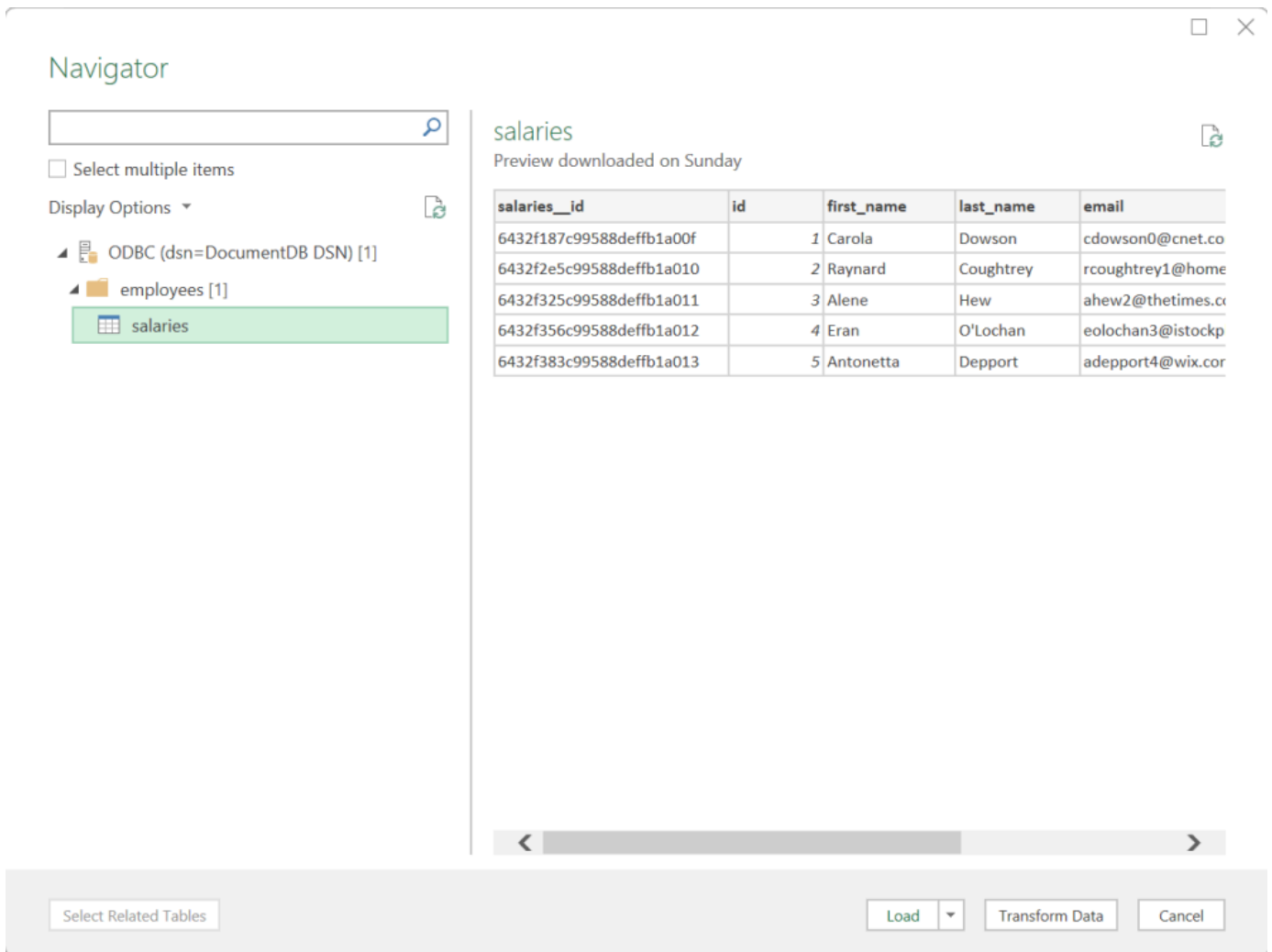
1. 確定已正確安裝和 Amazon DocumentDB 驅動程式。如需其他資訊，請參閱[在視窗中設定 ODBC 驅動程式](#)。
2. 啟動微軟 Excel。
3. 導覽至 [資料] > [取得資料] > [從其他來源]。
4. 從中選擇：



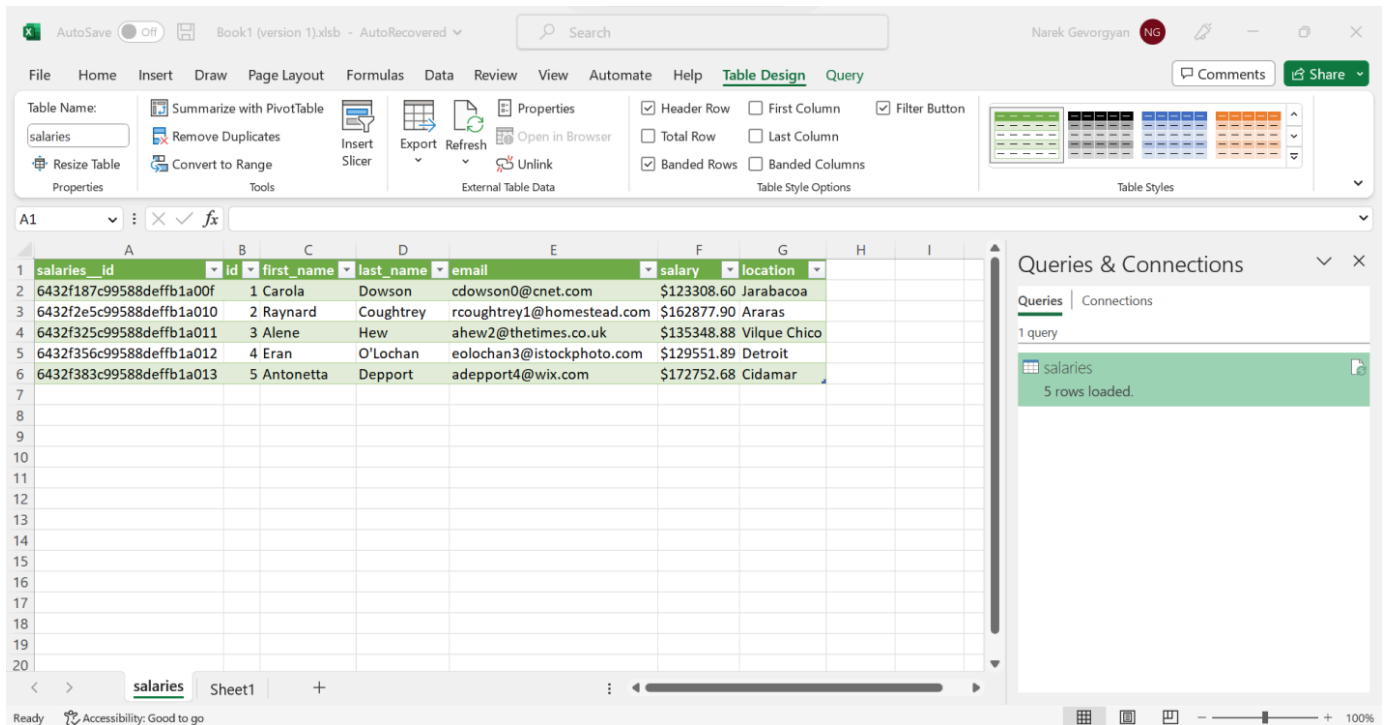
5. 從與 Amazon 文件資料庫關聯的資料來源名稱 (DSN) 下拉式功能表中選取資料來源：



6. 選擇您要從中載入資料的集合：



7. 將資料載入 Excel 中：



從微軟電力 BI 桌面 Connect 到 Amazon DocumentDB

主題

- [先決條件](#)
- [添加微軟電源 BI 桌面自定義連接器](#)
- [使用 Amazon DocumentDB 自訂連接器進行連線](#)
- [配置微軟電源 BI 網關](#)

先決條件

在開始之前，請確定已正確安裝 Amazon DocumentDB ODBC 驅動程式。

添加微軟電源 BI 桌面自定義連接器

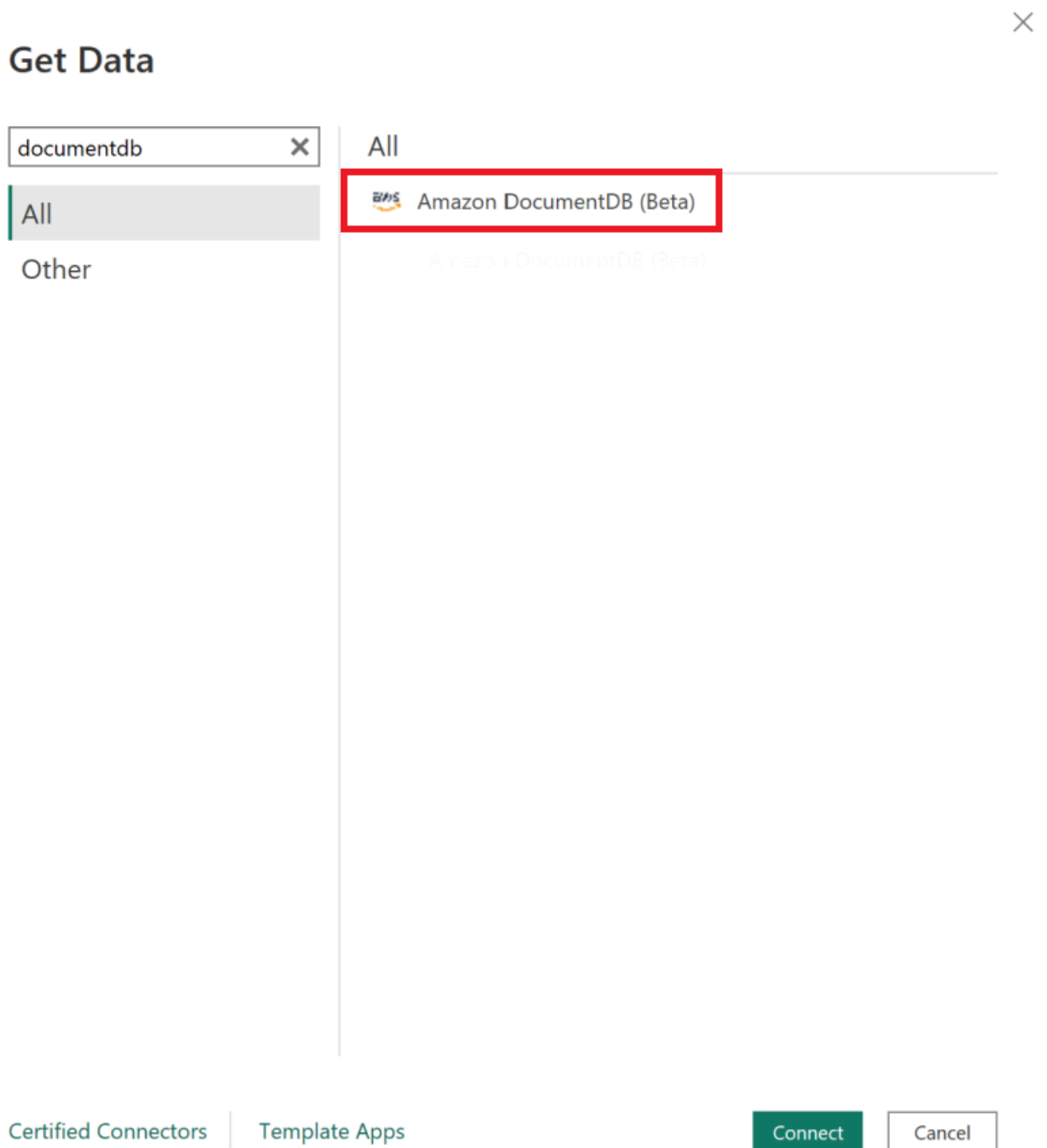
將AmazonDocumentDBConnector.mez文件複製到文件<User>\Documents\Power BI Desktop\Custom Connectors\夾 (<User>\OneDrive\Documents\Power BI Desktop\Custom Connectors如果使用，則將其複製到文件夾中 OneDrive)。這將允許 Power BI 訪問自定義連接器。您可以在[此處](#)獲取到 Power BI 桌面的連接器。重新啟動 Power BI 桌面，以確定連接器已載入。

Note

自訂連接器僅支援 Amazon DocumentDB 使用者名稱和密碼進行身份驗證。

使用 Amazon DocumentDB 自訂連接器進行連線

1. 從取得資料中選取 Amazon 文件資料庫 (測試版)，然後按一下 Connect。如果您收到使用第三方服務的警告，請按一下「繼續」。



2. 輸入連線到 Amazon DocumentDB 叢集的所有必要資訊，然後按一下確定：



Amazon DocumentDB

HostName ⓘ

Port ⓘ

Database ⓘ

TLS (optional) ⓘ

Allow Invalid HostNames (optional) ⓘ

TLS CA File Path (optional) ⓘ

Enable SSH tunnel (optional) ⓘ

SSH tunnel user (optional) ⓘ

SSH tunnel hostname (optional) ⓘ

SSH tunnel private certificate path (optional) ⓘ

OK

Cancel

Note

根據 ODBC 驅動程式之資料來源名稱 (DSN) 的組態，如果您已在 DSN 設定中提供必要的資訊，則 SSH 連線詳細資料畫面可能不會顯示。

3. 選擇資料連線模式：

- 導入-加載所有數據並將信息存儲在磁盤上。資料必須重新整理並重新載入，才能顯示資料更新。
- 直接查詢-不加載數據，但會對數據進行實時查詢。這表示不需要重新整理和重新載入資料，以顯示資料更新。

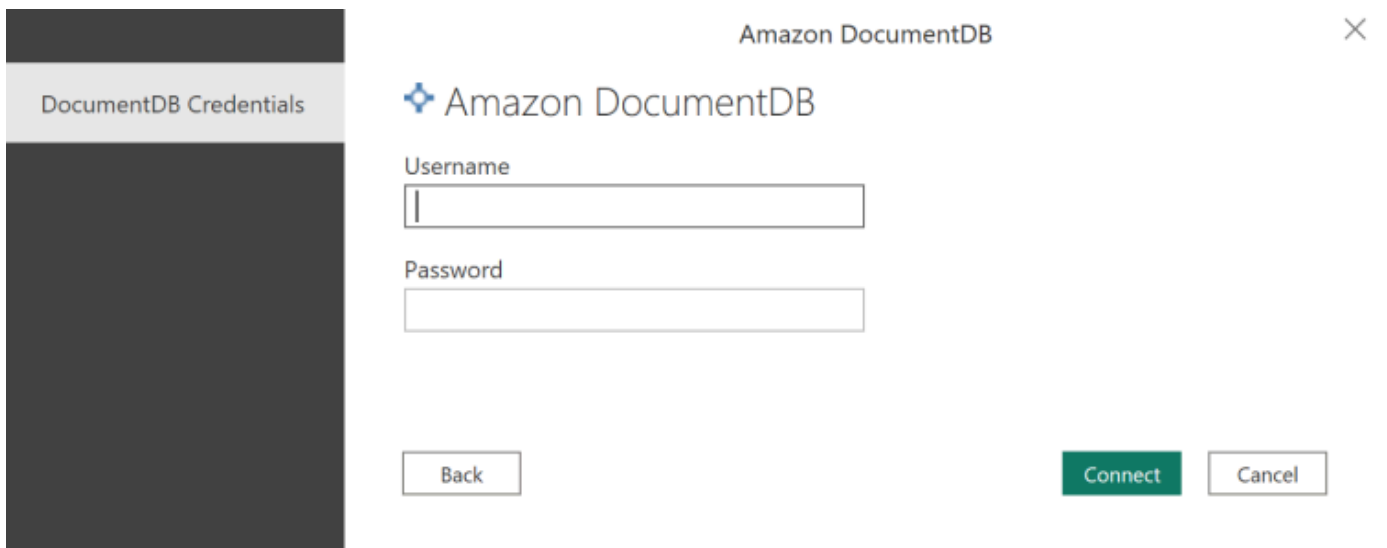


The screenshot shows a dialog box titled "Amazon DocumentDB" with a close button (X) in the top right corner. Inside the dialog, there is a section labeled "DSN" with a sub-label "DSN ⓘ" and a text input field containing "DocumentDB DSN". Below this is a section labeled "Data Connectivity mode ⓘ" with two radio button options: "Import" (which is selected) and "DirectQuery". At the bottom right of the dialog are two buttons: "OK" and "Cancel".

Note

如果您使用的是非常大的資料集，匯入所有資料可能需要較長的時間。

4. 如果這是第一次連線至此資料來源，請選取驗證類型並在出現提示時輸入您的認證。然後點擊 Connect：



The screenshot shows a dialog box titled "Amazon DocumentDB" with a close button (X) in the top right corner. On the left side, there is a dark sidebar with a tab labeled "DocumentDB Credentials". The main area of the dialog has the Amazon DocumentDB logo and the text "Amazon DocumentDB". Below this, there are two text input fields: "Username" and "Password". At the bottom of the dialog are three buttons: "Back", "Connect", and "Cancel".

5. 在「瀏覽」對話方塊中，選取所需的資料庫表格，然後按一下載入載入資料，或按一下轉換資料繼續轉換資料。

Navigator

The Navigator interface displays a tree view of data sources on the left and a table of query results on the right. The tree view shows a folder named 'odbc-test [20]' containing several sub-folders and files. The file 'queries_test_001' is selected. The table on the right shows the results of a query against 'queries_test_001'.

| queries_test_001_id | fieldDecimal128 | fieldDouble | fieldString | fieldObjectId |
|--------------------------|-----------------|--------------|-------------|--------------------------|
| 62196dcc4d91892191475139 | 3.40282E+20 | 1.79769E+308 | some Text | 62196dcc4d91892191475139 |

Buttons at the bottom right: Load, Transform Data, Cancel.

Note

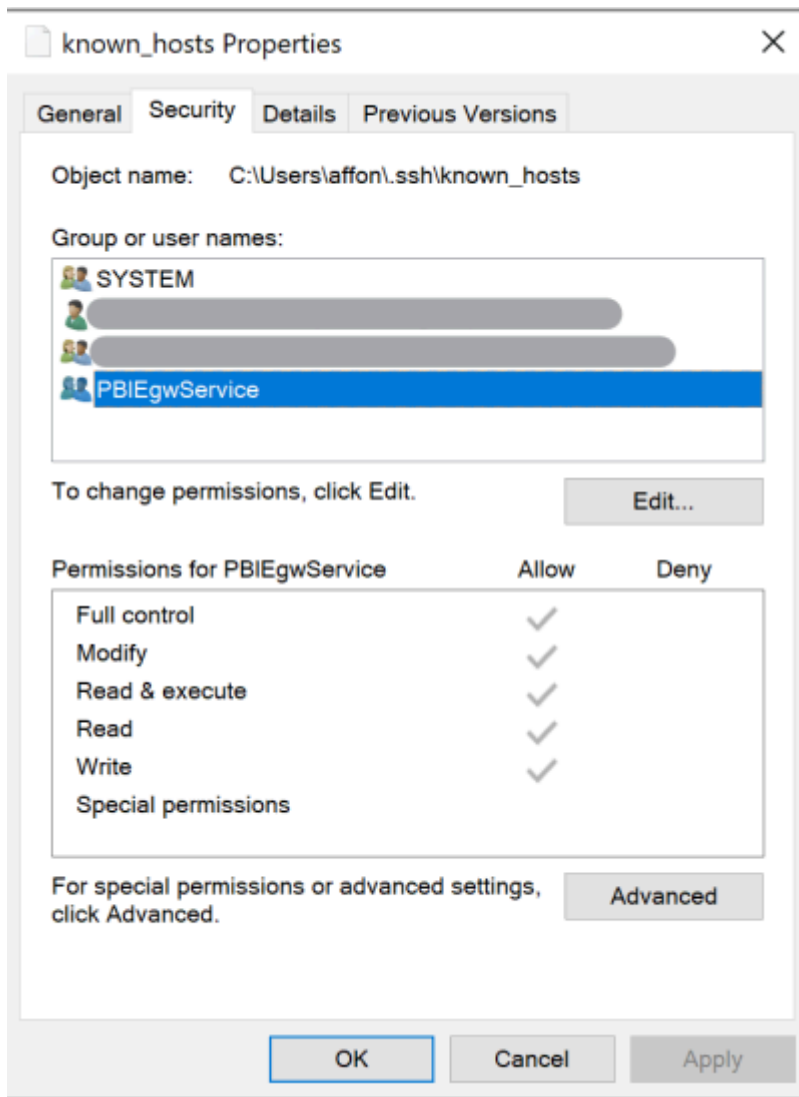
連線後，即會儲存資料來源設定。若要修改它們，請選取「轉換資料 > 資料來源設定」。

配置微軟電源 BI 網關

先決條件：

- 確保自定義連接器將與電源 BI 網關一起工作。
- 請確定 ODBC DSN 是在安裝 Power BI 閘道的機器上的 [系統] 索引標籤的 ODBC 資料來源中建立。

如果您使用的是內部 SSH 通道功能，則該檔案必known_hosts須位於 Power BI 服務帳戶可存取該檔案的位置。



Note

這也適用於您可能需要能夠與 Amazon DocumentDB 叢集建立連線的任何檔案，例如憑證授權單位 (CA) 憑證檔案 (pem 檔案)。

自動生成模式

ODBC 驅動程序通過 JNI (Java 本機界面) 利用 Amazon DocumentDB JDBC 驅動程序-使自動模式生成功能在 JDBC 驅動程序中的工作方式類似。如需自動產生結構描述的詳細資訊，請參閱 [JDBC 自動產生結構描述](#)。此外，要了解有關 ODBC 驅動程序體系結構的更多信息，請單擊[此處](#)。

SQL 支援和限制

Amazon DocumentDB ODBC 驅動程序是一個只讀驅動程序，支持 SQL-92 的子集和一些常見的擴展。如需詳細資訊，請參閱 [ODBC 支援和限制](#) 文件。

疑難排解

如果您在使用 Amazon DocumentDB ODBC 驅動程式時遇到問題，請參閱 [疑難排解指南](#)。

Amazon DocumentDB 配額和限制

本主題說明 Amazon 文件資料庫 (與 MongoDB 相容性) 的資源配額、限制和命名限制。

對於某些管理功能，Amazon DocumentDB 會使用與 Amazon Relational Database Service 服務 (Amazon RDS) 和亞馬 Amazon Neptune 共用的操作技術。

主題

- [支援的執行個體類型](#)
- [支援地區](#)
- [區域配額](#)
- [彙總限制](#)
- [叢集限制](#)
- [執行個體限制](#)
- [命名限制](#)
- [TTL 限制](#)
- [彈性叢集限制](#)
- [彈性叢集碎片限制](#)
- [每個碎片的彈性叢集 CPU、記憶體、連線和游標限制](#)

支援的執行個體類型

Amazon DocumentDB 支援隨需執行個體和下列執行個體類型：

- 記憶體最佳化：
 - R6G 執行個體類
型：db.r6g.large、db.r6g.2xlarge、db.r6g.4xlarge、db.r6g.8xlarge、db.r6g.12xlarge
 - R5 執行個體類
型：db.r5.large、db.r5.2xlarge、db.r5.4xlarge、db.r5.8xlarge、db.r5.12xlarge
 - R4 執行個體類
型：db.r4.large、db.r4.2xlarge、db.r4.4xlarge、db.r4.8xlarge、db.r4.16xlarge
- 高載性能：
 - T4G 執行個體類型：db.t4g.medium

- T3 執行個體類型：db.t3.medium。

如需有關支援的執行個體類型及其規格的詳細資訊，請參閱[執行個體類別規格](#)。

支援地區

Amazon DocumentDB 在以下區域提供：AWS

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------|----------------|-----------|
| 美國東部 (俄亥俄) | us-east-2 | 3 |
| 美國東部 (維吉尼亞北部) | us-east-1 | 6 |
| 美國西部 (奧勒岡) | us-west-2 | 4 |
| 南美洲 (聖保羅) | sa-east-1 | 3 |
| 亞太區域 (香港) | ap-east-1 | 3 |
| 亞太區域 (海德拉巴) | ap-south-2 | 3 |
| 亞太區域 (孟買) | ap-south-1 | 3 |
| 亞太區域 (首爾) | ap-northeast-2 | 4 |
| 亞太區域 (新加坡) | ap-southeast-1 | 3 |
| 亞太區域 (悉尼) | ap-southeast-2 | 3 |
| 亞太區域 (東京) | ap-northeast-1 | 3 |
| 加拿大 (中部) | ca-central-1 | 3 |
| 中國 (北京) 區域 | cn-north-1 | 3 |
| 中國 (寧夏) | cn-northwest-1 | 3 |
| 歐洲 (法蘭克福) | eu-central-1 | 3 |
| 歐洲 (愛爾蘭) | eu-west-1 | 3 |
| 歐洲 (倫敦) | eu-west-2 | 3 |

| 區域名稱 | 區域 | 可用區域 (計算) |
|---------------------|---------------|-----------|
| 歐洲 (米蘭) | eu-south-1 | 3 |
| Europe (Paris) | eu-west-3 | 3 |
| 中東 (阿拉伯聯合大公國) | me-central-1 | 3 |
| AWS GovCloud (美國西部) | us-gov-west-1 | 3 |
| AWS GovCloud (美國東部) | us-gov-east-1 | 3 |

區域配額

對於某些管理功能，Amazon DocumentDB 使用與 Amazon Relational Database Service (Amazon RDS) 共享的操作技術。下表包含 Amazon DocumentDB 和 Amazon RDS 之間共享的區域限制。

Note

上述 Amazon RDS 共用技術僅適用於以 Amazon DocumentDB 執行個體為基礎的叢集。Amazon DocumentDB 彈性群集不與 Amazon RDS 共享技術。

以下限制適用於以 Amazon DocumentDB 執行個體為基礎的叢集，而且是每個區域的每個 AWS 帳戶。

| 資源 | AWS 預設限制 |
|--------|----------|
| 叢集 | 40 |
| 叢集參數群組 | 50 |
| 事件訂閱 | 20 |
| 執行個體 | 40 |

| 資源 | AWS 預設限制 |
|------------------|----------|
| 手動叢集快照 | 100 |
| 僅供讀取複本數 (每個叢集) | 15 |
| 子網路群組 | 50 |
| 每個子網路群組的子網路 | 20 |
| 每個資源的標籤 | 50 |
| 每個執行個體的 VPC 安全群組 | 5 |

下列限制適用於 Amazon DocumentDB 彈性叢集，而且是每個區域的每個 AWS 帳戶。

| 資源 | AWS 預設限制 |
|-----------|----------|
| 彈性叢集 | 20 |
| 彈性叢集 vCPU | 1024 |
| 手動彈性叢集快照 | 20 |

如果配額可調整，您可以使用 Service Quotas 要求增加配額。有些要求會自動解決，而其他要求則會提交至 AWS Support。您可以追蹤提交的配額增加要求的狀態 AWS Support。增加服務配額的請求不會獲得優先支援。如果您有緊急要求，請聯繫 [AWS Support](#)。如需服務配額的詳細資訊，請參閱 [什麼是服務配額？](#)

要求增加 Amazon DocumentDB 的配額：

1. 在 <https://console.aws.amazon.com/servicequotas> 開啟 Service Quotas (服務配額) 主控台，如有需要，請登入。
2. 在導覽窗格中，選擇 AWS services (AWS 服務)。
3. 從清單中選取 Amazon DocumentDB 料庫 (與 MongoDB 相容性) 或 Amazon DocumentDB 彈性叢集，或在搜尋欄位中輸入任一項。

4. 如果配額可調整，您可以選取其選項按鈕或名稱，然後選擇頁面右上角的 Request quota increase (請求增加配額)。
5. 對於 Change quota value (變更配額值)，輸入新值。新值必須大於目前的值。
6. 選擇請求。解決請求之後，配額的 Applied quota value (套用的配額值) 會設定為新值。
7. 若要檢視任何擱置或最近解決的請求，請從瀏覽窗格中選擇 Dashboard (儀表板)。對於擱置的請求，請選擇請求狀態以開啟請求回條。請求的初始狀態為 Pending。狀態變更為後 Quota requested，您會看到案例編號與 AWS Support。選擇案例編號，為請求開啟票證。

彙總限制

下表說明 Amazon DocumentDB 中的彙總限制。

| 資源 | 限制 |
|----------|-----|
| 支援之階段數上限 | 500 |

叢集限制

下表說明 Amazon DocumentDB 執行個體型叢集限制。

| 資源 | 限制 |
|----------------------------------|-----------|
| 叢集大小 (所有集合和索引的總和) | 128 TiB |
| 集合大小 (所有集合的總和不得超過叢集限制) - 不包含索引大小 | 32 TB |
| 每個叢集的集合 | 100,000 |
| 每個叢集的資料庫數量 | 100,000 |
| 資料庫大小 (所有資料庫的總和不得超過叢集限制) | 128 TiB |
| 文件巢狀深度 | 200 個等級 |
| 文件大小 | 16 MB |
| 索引鍵大小 | 2048 個位元組 |
| 每個集合的索引 | 64 |
| 在複合索引的索引鍵 | 32 |
| 單一批次命令中的寫入次數上限 | 100,000 |
| 每個叢集的使用者數目 | 1000 |

執行個體限制

下表說明每個執行個體的 Amazon DocumentDB 限制。

| 執行個體類型 | 執行個體記憶體 (GiB) | 連接 (全部) | 游標限制 | 未結交易 | 連線 (作用中) |
|--------|---------------|---------|------|------|----------|
| T3. 中等 | 4 | 500 | 30 | 50 | 102 |
| T4 克。中 | 4 | 500 | 30 | 50 | 102 |
| R4. 大 | 15.25 | 1700 | 450 | N/A | 1100 |
| 大 | 30.5 | 3400 | 450 | N/A | 2700 |
| 大 | 61 | 6800 | 450 | N/A | 4500 |
| 大 | 122 | 13600 | 725 | N/A | 4500 |
| 大 | 288 | 27200 | 1450 | N/A | 4500 |
| 大型 | 488 | 30000 | 2900 | N/A | 4500 |
| 大 | 16 | 1700 | 450 | 200 | 1100 |
| 大 | 32 | 3500 | 450 | 400 | 2700 |
| 大型 | 64 | 7100 | 450 | 800 | 4500 |
| 大型 | 128 | 14200 | 760 | 1600 | 4500 |
| 大型 | 256 | 28400 | 1520 | 3200 | 4500 |
| 大型 | 383 | 30000 | 2280 | 4800 | 4500 |
| 大型 | 512 | 30000 | 3040 | 6400 | 4500 |
| 大型 | 768 | 30000 | 4560 | 9600 | 4500 |
| 大 | 16 | 1700 | 450 | 200 | 1100 |

| 執行個體類型 | 執行個體記憶體 (GiB) | 連接 (全部) | 游標限制 | 未結交易 | 連線 (作用中) |
|--------|---------------|---------|------|------|----------|
| 大尺寸 | 32 | 3500 | 450 | 400 | 2700 |
| 大号 | 64 | 7100 | 450 | 800 | 4500 |
| 大号 | 128 | 14200 | 760 | 1600 | 4500 |
| 大号 | 256 | 28400 | 1520 | 3200 | 4500 |
| 大号 | 383 | 30000 | 2280 | 4800 | 4500 |
| 大号 | 512 | 30000 | 3040 | 6400 | 4500 |

您可以使用下列 CloudWatch 指標來監控每個執行個體限制並發出警示。如需 Amazon 文件資料庫 CloudWatch 指標的詳細資訊，請參閱。[使用監控亞馬遜文檔數據庫 CloudWatch](#)

| 限制 | CloudWatch 指標 |
|------|------------------------|
| 實例內存 | FreeableMemory |
| 連線 | DatabaseConnectionsMax |
| 游標 | DatabaseCursorsMax |
| 交易 | TransactionsOpenMax |

命名限制

下表說明 Amazon DocumentDB 中的命名限制。

| 資源 | 預設限制 |
|-------------------------------|---|
| 叢集識別碼 | <ul style="list-style-type: none"> 長度為 [1—63] 字母、數字或連字號。 第一個字元必須是字母。 不能以連字號結尾或連續包含兩個連字號。 每個 AWS 帳戶、每個區域的所有叢集 (跨 Amazon RDS、亞馬遜海王星和 Amazon DocumentDB) 必須是唯一的。 |
| 集合名稱 : <col> | 長度為 [1—57] 個字元。 |
| 資料庫名稱 : <db> | 長度為 [1—63] 個字元。 |
| 完整集合名稱 : <db>.<col> | 長度為 [3—120] 個字元。 |
| 完整索引名稱 : <db>.<col>.\$<index> | 長度為 [6—127] 個字元。 |
| 索引名稱 : <col>\$<index> | 長度為 [3—63] 個字元。 |
| 執行個體識別碼 | <ul style="list-style-type: none"> 長度為 [1—63] 字母、數字或連字號 第一個字元必須是字母 不能以連字號結尾或連續包含兩個連字號 每個 AWS 帳戶、每個區域的所有執行個體 (跨 Amazon RDS、亞馬遜海王星和 Amazon DocumentDB) 必須是唯一的。 |

| 資源 | 預設限制 |
|------------------------|---|
| Master password (主要密碼) | <ul style="list-style-type: none"> • 長度為 [8-100] 個可列印的 ASCII 字元。 • 可以使用下列項目以外的任何可列印 ASCII 字元： <ul style="list-style-type: none"> • / (正斜線) • " (雙引號) • @ (@ 符號) |
| 主要使用者名稱 | <ul style="list-style-type: none"> • 長度為 [1-63] 英數字元。 • 第一個字元必須是字母。 • 不能是資料庫引擎保留的字。 |
| 參數群組名稱 | <ul style="list-style-type: none"> • 長度為 [1-255] 個英數字元。 • 第一個字元必須是字母。 • 不能以連字號結尾或連續包含兩個連字號。 |

TTL 限制

在特定時間範圍內無法保證可從 TTL 索引中刪除，而且這是最佳作法。像是執行個體資源使用率、文件大小和整體輸送量等因素，都會影響 TTL 刪除的時間。

彈性叢集限制

下表說明 Amazon DocumentDB 彈性叢集中的最大限制。

| 資源 | 限制 |
|--------------------|------|
| 每個區域的彈性叢集 | 20 |
| 每個區域的所有彈性叢集加總 vCPU | 1024 |
| 每個區域的手動叢集快照 | 20 |

| 資源 | 限制 |
|---------------------------|---|
| 每個叢集的碎片 | 32 |
| 每個叢集的儲存空間 (當資料以銳利金鑰平均分配時) | 4 PIB |
| 叢集的連線 | 每個碎片的值下限為 300,000 或碎片數目 x 與 vCPU 相關聯的連線限制 |
| UnSharded 集合大小 | 32 TB |
| 分片集合大小 (當數據由銳鍵均勻分佈時) | 1 PB |
| 每個叢集的資料庫數量 | 10,000 |
| UnSharded 每個叢集的集合 | 100,000 |
| 每個叢集的分割集合 | 1000 |
| 每個叢集使用者 | 100 |
| 在單個批處理命令中寫入 | 100,000 |
| 每個集合的索引 | 64 |
| 文件巢狀深度 | 100 個層級 |
| 文件大小 | 16MB |
| 索引鍵大小 | 2048 位元組 |
| 在複合索引的索引鍵 | 32 |

彈性叢集碎片限制

下表說明 Amazon DocumentDB 彈性叢集中的最大碎片限制。

| 資源 | 限制 |
|-------------------|---------|
| 每個碎片執行個體的 vCPU 數量 | 64 |
| 每個碎片的執行個體 | 16 |
| 每個碎片的排量 | 128 TiB |
| 每個集合每個碎片的儲存 | 32 TB |

每個碎片的彈性叢集 CPU、記憶體、連線和游標限制

下表說明 Amazon DocumentDB 彈性叢集碎片中的最大 CPU、記憶體、連線和游標限制。

| 每個碎片的 vCPUs 數量 | 執行個體記憶體 (GiB) | 連線限制 | 游標限制 |
|----------------|---------------|-------|------|
| 2 | 16 | 1700 | 450 |
| 4 | 32 | 3500 | 450 |
| 8 | 64 | 7100 | 450 |
| 16 | 128 | 14200 | 760 |
| 32 | 256 | 28400 | 1520 |
| 48 | 383 | 30000 | 2280 |
| 64 | 512 | 30000 | 3040 |

查詢

本節說明使用 Amazon DocumentDB 查詢的所有方面。

主題

- [查詢文件](#)
- [查詢計劃](#)
- [解釋結果](#)
- [使用 Amazon DocumentDB 數據查詢地理空間數據](#)
- [部分索引](#)
- [使用 Amazon DocumentDB 執行文本搜索](#)

查詢文件

有時候，您可能需要查詢線上商店的庫存，讓客戶可以查看和購買您銷售的商品。查詢集合相當容易，無論您需要的是集合中的所有文件，或僅需要滿足特定條件的文件。

若要查詢文件，請使用 `find()` 操作。`find()` 命令擁有單一文件參數，負責定義選擇要傳回的文件時所使用的條件。`find()` 輸出的文件格式會是單一行文字且沒有換行。若要將輸出文件格式化，以更方便閱讀，請使用 `find().pretty()`。這個主題的所有範例都使用 `.pretty()` 設定輸出格式。

下列程式碼範例會使用您在前兩個練習中插入到 `example` 集合中的四個文件，`insertOne()` 而 `insertMany()` 這些文件位於「[使用文件](#)」的「[新增文件](#)」區段中。

主題

- [檢索集合中的所有文檔](#)
- [檢索匹配字段值的文檔](#)
- [檢索與嵌入文檔匹配的文檔](#)
- [檢索與嵌入文檔中的字段值匹配的文檔](#)
- [檢索匹配數組的文檔](#)
- [檢索匹配數組中的值的文檔](#)
- [使用運算子擷取文件](#)

檢索集合中的所有文檔

若要擷取集合中的所有文件，請使用空的查詢文件使用 `find()` 操作。

以下查詢會傳回 `example` 集合中的所有文件。

```
db.example.find( {} ).pretty()
```

檢索匹配字段值的文檔

若要擷取符合某個欄位和值的所有文件，請使用識別要符合的欄位和值的查詢文件使用 `find()` 操作。

使用先前文件，此查詢會傳回所有文件，其中「項目」欄位等於「筆」。

```
db.example.find( { "Item": "Pen" } ).pretty()
```

檢索與嵌入文檔匹配的文檔

若要尋找符合嵌入文件的所有文件，請使用 `find()` 操作，其附帶的查詢文件會指定內嵌文件名稱和該內嵌文件的所有欄位和值。

當符合嵌入文件，文件的內嵌文件必須擁有與查詢相同的名稱。此外，內嵌文件的欄位和值必須符合查詢。

以下查詢只會傳回「Poster Paint」(廣告顏料) 文件。這是因為「Pen」(筆) 的 `OnHand` 和 `MinOnHand` 有不同值，而「Spray Paint」(噴漆) 比查詢文件多一個欄位 (`OrderQty`)。

```
db.example.find({"Inventory": {  
  "OnHand": 47,  
  "MinOnHand": 50 } } ).pretty()
```

檢索與嵌入文檔中的字段值匹配的文檔

若要尋找符合嵌入文件的所有文件，請使用 `find()` 操作，其附帶的查詢文件會指定內嵌文件名稱和該內嵌文件的所有欄位和值。

鑒於前述文件，以下查詢使用「點表示法」指定內嵌文件和關注欄位。符合這些的任何文件將會傳回，無論內嵌文件中可能有哪些其他欄位。查詢將傳回「廣告顏料」和「噴漆」，因為它們都符合指定的欄位和值。

```
db.example.find({"Inventory.OnHand": 47, "Inventory.MinOnHand": 50 }).pretty()
```

檢索匹配數組的文檔

若要尋找符合陣列的所有文件，請使用 `find()` 操作與您感興趣的陣列名稱，以及在該陣列中的所有值。查詢將傳回所有具有該名稱的文件，其中陣列值與查詢中值及順序相同。

以下查詢只會傳回「筆」，因為「廣告顏料」有額外的顏色（白色），而「噴漆」有不同順序的顏色。

```
db.example.find( { "Colors": ["Red","Green","Blue","Black"] } ).pretty()
```

檢索匹配數組中的值的文檔

若要尋找具有特殊陣列值的所有文件，請使用 `find()` 操作與您感興趣的陣列名稱和值。

```
db.example.find( { "Colors": "Red" } ).pretty()
```

前述操作會傳回全部三個文件，因為每個文件都有一個名為 `Colors` 的陣列，且陣列中某處具有「Red」值。如果您指定數值「White」，查詢只會傳回「廣告顏料」。

使用運算子擷取文件

以下查詢會傳回其中「Inventory.OnHand」值小於 50 的所有文件。

```
db.example.find(
  { "Inventory.OnHand": { $lt: 50 } } )
```

如需支援的查詢運算子清單的詳細資訊，請參閱[查詢和投影運算子](#)。

查詢計劃

如何查看查詢計劃的 `executionStats` ？

判斷查詢執行速度比預期慢的原因時，了解查詢計劃的 `executionStats` 可能很有幫助。`executionStats` 提供從特定階段傳回的文件數量 (`nReturned`)、在每個階段花費的執行時間量 (`executionTimeMillisEstimate`)，以及產生查詢計劃 (`planningTimeMillis`) 所需的時間量。您可以從 `executionStats` 輸出中判斷查詢中最耗時的階段，以協助專注於最佳化工作，如下列查詢範例所示。`executionStats` 參數目前不支援 `update` 和 `delete` 命令。

Note

Amazon DocumentDB 會在使用分散式容錯、自我修復儲存系統的專用資料庫引擎上模擬 MongoDB 3.6 API。因此，Amazon DocumentDB 和 MongoDB 之間的查詢計劃和輸出 `explain()` 可能會有所不同。想要控制其查詢計劃的客戶可以使用 `$hint` 運算子強制選取偏好的索引。

請在 `explain()` 命令下執行您要改善的查詢，如下所示。

```
db.runCommand({explain: {query document}}).  
explain("executionStats").executionStats;
```

下面是範例操作。

```
db.fish.find({}).limit(2).explain("executionStats");
```

此操作的輸出將會如下所示。

```
{  
  "queryPlanner" : {  
    "plannerVersion" : 1,  
    "namespace" : "test.fish",  
    "winningPlan" : {  
      "stage" : "SUBSCAN",  
      "inputStage" : {  
        "stage" : "LIMIT_SKIP",  
        "inputStage" : {  
          "stage" : "COLLSCAN"  
        }  
      }  
    }  
  },  
  "executionStats" : {  
    "executionSuccess" : true,  
    "executionTimeMillis" : "0.063",  
    "planningTimeMillis" : "0.040",  
    "executionStages" : {  
      "stage" : "SUBSCAN",  
      "nReturned" : "2",  
      "executionTimeMillisEstimate" : "0.012",
```

```
    "inputStage" : {
      "stage" : "LIMIT_SKIP",
      "nReturned" : "2",
      "executionTimeMillisEstimate" : "0.005",
      "inputStage" : {
        "stage" : "COLLSCAN",
        "nReturned" : "2",
        "executionTimeMillisEstimate" : "0.005"
      }
    }
  },
  "serverInfo" : {
    "host" : "enginedemo",
    "port" : 27017,
    "version" : "3.6.0"
  },
  "ok" : 1
}
```

如果您只有興趣從上面的查詢查看 `executionStats`，可以使用下面的命令。對於小型集合，如果效能提升可忽略不計，Amazon DocumentDB 查詢處理器可以選擇不使用索引。

```
db.fish.find({}).limit(2).explain("executionStats").executionStats;
```

查詢計畫快取

為了最佳化效能並縮短規劃持續時間，Amazon DocumentDB 會在內部快取查詢計畫。這可讓使用快取計畫直接執行具有相同形狀的查詢。

不過，這種快取有時可能會造成相同查詢的隨機延遲；例如，通常需要一秒鐘才能執行的查詢偶爾需要十秒鐘的時間。這是因為隨著時間的推移，`reader` 實例緩存了查詢的各種形狀，從而消耗內存。如果您遇到這種隨機緩慢的情況，則不需要執行任何動作來釋放內存-系統將為您管理內存使用情況，並且一旦內存達到特定閾值，它將自動釋放。

解釋結果

如果您想要傳回查詢計畫的相關資訊，Amazon DocumentDB 支援詳細資訊模式。`queryPlannerexplain`結果會以類似下列的格式傳回最佳化處理程式所選取的查詢計畫：

```
{
  "queryPlanner" : {
    "plannerVersion" : <int>,
    "namespace" : <string>,
    "winningPlan" : {
      "stage" : <STAGE1>,
      ...
      "inputStage" : {
        "stage" : <STAGE2>,
        ...
        "inputStage" : {
          ...
        }
      }
    }
  }
}
```

以下各節將定義共同的explain結果。

主題

- [掃描和篩選階段](#)
- [索引交集](#)
- [索引聯集](#)
- [多個索引交點/聯集](#)
- [化合物索引](#)
- [排序階段](#)
- [小組賽階段](#)

掃描和篩選階段

最佳化程式可以選擇下列其中一項掃描：

科爾斯掃描

此階段為連續收集掃描。

```
{
```

```
"stage" : "COLLSCAN"
}
```

伊斯掃描

此階段掃描索引鍵。最佳化工具可能會擷取此階段內的文件，這可能會導致稍後附加的 FETCH 階段。

```
db.foo.find({"a": 1})
{
  "stage" : "IXSCAN",
  "direction" : "forward",
  "indexName" : <idx_name>
}
```

FETCH

如果最佳化程式擷取了 IXSCAN 以外的階段中的文件，則結果將包含 FETCH 階段。例如，上面的 IXSCAN 查詢可能會導致擷取和 IXSCAN 階段的組合：

```
db.foo.find({"a": 1})
{
  "stage" : "FETCH",
  "inputStage" : {
    "stage" : "IXSCAN",
    "indexName" : <idx_name>
  }
}
```

IXONLYSCAN 僅掃描索引鍵。創建複合索引不會避免 FETCH。

索引交集

IXAND

如果 Amazon DocumentDB 可以利用索引交集，則亞馬遜文檔數據庫可以包含帶有 IXSCAN 輸入階段數組的 IXAND 階段。例如，我們可能會看到如下輸出：

```
{
```

```

"stage" : "FETCH",
"inputStage" : {
  "stage" : "IXAND",
  "inputStages" : [
    {
      "stage" : "IXSCAN",
      "indexName" : "a_1"
    },
    {
      "stage" : "IXSCAN",
      "indexName" : "b_1"
    }
  ]
}
}

```

索引聯集

IXOR

類似於索引交集，Amazon DocumentDB 可以包括IXOR階段與運營商的inputStages陣列。\$or

```
db.foo.find({"$or": [{"a": {"$gt": 2}}, {"b": {"$lt": 2}}]})
```

對於上面的查詢，解釋輸出可能如下所示：

```

{
  "stage" : "FETCH",
  "inputStage" : {
    "stage" : "IXOR",
    "inputStages" : [
      {
        "stage" : "IXSCAN",
        "indexName" : "a_1"
      },
      {
        "stage" : "IXSCAN",
        "indexName" : "b_1"
      }
    ]
  }
}

```

```
}

```

多個索引交點/聯集

Amazon DocumentDB 可以將多個索引交集或聯集階段組合在一起，然後獲取結果。例如：

```
{
  "stage" : "FETCH",
  "inputStage" : {
    "stage" : "IXOR",
    "inputStages" : [
      {
        "stage" : "IXSCAN",
        ...
      },
      {
        "stage" : "IXAND",
        "inputStages" : [
          {
            "stage" : "IXSCAN",
            ...
          },
          {
            "stage" : "IXSCAN",
            ...
          }
        ]
      }
    ]
  }
}

```

索引交集或聯集階段的使用不受索引類型 (稀疏、複合等) 的影響。

化合物索引

Amazon DocumentDB 複合索引使用不受索引欄位開始子集的限制；它可以使用索引與尾碼部分，但效率可能不太高。

例如，的複合索引 { a: 1, b: -1 } 可以支援以下所有三個查詢：


```
db.orders.find( { a: 1 } )
```

```
db.orders.find( { b: 1 } )
```

```
db.orders.find( { a: 1, b: 1 } )
```

排序階段

如果要求的排序索引鍵上有索引，Amazon DocumentDB 可以使用索引來取得順序。在這種情況下，結果將不包括SORT階段，而是一個IXSCAN階段。如果優化器喜歡簡單的排序，它將包含這樣的階段：

```
{
  "stage" : "SORT",
  "sortPattern" : {
    "a" : 1,
    "b" : -1
  }
}
```

小組賽階段

Amazon DocumentDB 支持兩種不同的組策略：

- SORT_AGGREGATE：在磁盤上排序聚合。
- HASH_AGGREGATE: 在記憶體雜湊彙總中。

使用 Amazon DocumentDB 數據查詢地理空間數據

本節介紹如何使用 Amazon DocumentDB 查詢地理空間資料。閱讀本節之後，您將能夠回答如何在 Amazon DocumentDB 中存儲，查詢和索引地理空間數據。

主題

- [概要](#)
- [索引和存儲地理空間數據](#)
- [查詢地理空間資料](#)
- [限制](#)

概要

地理空間的常見使用案例涉及來自資料的接近分析。例如，「尋找西雅圖 50 英里範圍內的所有機場」，或「尋找指定位置最近的餐廳」。Amazon DocumentDB 使用 [GeoJSON 規範](#) 來表示地理空間數據。GeoJSON 是用於坐標空間中形狀的 JSON 格式的開源規範。GeoJSON 座標可擷取經度和緯度，代表地球狀球體上的位置。

索引和存儲地理空間數據

Amazon DocumentDB 使用「點」GeoJSON 類型來存儲地理空間數據。每個 GeoJSON 文檔 (或子文檔) 通常由兩個字段組成：

- 類型-所表示的形狀，它通知 Amazon DocumentDB 如何解釋「坐標」字段。目前，Amazon DocumentDB 僅支持點
- 坐標-以數組中的對象表示的緯度和經度對-[經度，緯度]

Amazon DocumentDB 還使用 2dsphere 索引來索引地理空間數據。Amazon DocumentDB 支持索引點。Amazon DocumentDB 支持使用 2dsphere 索引進行近接查詢。

讓我們考慮一個場景，您正在構建食品外送服務的應用程序。你想存儲各種餐廳的緯度和經度對 Amazon DocumentDB。若要這麼做，首先建議您在「地理空間」欄位上建立索引，以保存緯度和經度對。

```
use restaurantsdb
db.usarestaurants.createIndex({location:"2dsphere"})
```

這個命令的輸出看起來像這樣：

```
{
  "createdCollectionAutomatically" : true,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

一旦你已經創建了一個索引，你可以開始插入數據到你的 Amazon DocumentDB 集合。

```
db.usarestaurants.insert({
  "state":"Washington",
  "city":"Seattle",
```

```
"name":"Thai Palace",
"rating": 4.8,
"location":{"
  "type":"Point",
  "coordinates":[
    -122.3264,
    47.6009
  ]
}
});

db.usarestaurants.insert({
  "state":"Washington",
  "city":"Seattle",
  "name":"Noodle House",
  "rating": 4.8,
  "location":{"
    "type":"Point",
    "coordinates":[
      -122.3517,
      47.6159
    ]
  }
});

db.usarestaurants.insert({
  "state":"Washington",
  "city":"Seattle",
  "name":"Curry House",
  "rating": 4.8,
  "location":{"
    "type":"Point",
    "coordinates":[
      -121.4517,
      47.6229
    ]
  }
});
```

查詢地理空間資料

Amazon DocumentDB 支援地理空間資料的鄰近、包含和交集查詢。鄰近查詢的一個很好的例子是找到所有距離小於某個距離且距離多於另一個點（城市）的點（所有機場）。包含查詢的一個很好的例子

是找到位於指定區域/多邊形 (紐約州) 中的所有點 (所有機場)。交叉查詢的一個很好的例子是找到與點 (城市) 相交的多邊形 (州)。您可以使用以下地理空間運營商從數據中獲得見解。

- **\$nearSphere**-\$nearSphere 是一個查找運算符，支持從最近到最遠的 GeoJSON 點查找點。
- **\$geoNear**-\$geoNear 是一個聚合運算符，支持計算從 GeoJSON 點以米為單位的距離。
- **\$minDistance**-\$minDistance 是一個查找運算符，與\$nearSphere或\$geoNear一起使用，用於過濾距中心點至少指定最小距離的文檔。
- **\$maxDistance**-\$maxDistance 是與\$nearSphere或一起使用的查找運算符，\$geoNear用於過濾距中心點指定最大距離的文檔。
- **\$geoWithin**-\$geoWithin 是一個查找運算符，支持查找具有完全存在於指定形狀 (如多邊形) 中的地理空間數據的文檔。
- **\$geoIntersects**-\$geoIntersects 是尋找運算子，支援尋找其地理空間資料與指定 GeoJSON 物件相交的文件。

Note

\$geoNear並在您的鄰近查詢中使用的 GeoJSON 字段上\$nearSphere需要一個 2dsphere 索引。

範例 1

在此示例中，您將學習如何找到從地址 (點) 最近的距離排序的所有餐廳 (點)。

若要執行此類查詢，您可以使\$geoNear用計算與另一個點之間的點集合距離。您也可以加入以測量distanceMultiplier以公里為單位的距離。

```
db.usarestaurants.aggregate([
  {
    "$geoNear": {
      "near": {
        "type": "Point",
        "coordinates": [
          -122.3516,
          47.6156
        ]
      }
    }
  },
```

```
        "spherical":true,
        "distanceField":"DistanceKilometers",
        "distanceMultiplier":0.001
    }
}
])
```

上面的命令將返回按照距指定點的距離（最接近最遠）排序的餐廳。這個命令的輸出看起來像這樣

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "state" : "Washington", "city" :
  "Seattle", "name" : "Noodle House", "rating" : 4.8, "location" : { "type" : "Point",
  "coordinates" : [ -122.3517, 47.6159 ] }, "DistanceKilometers" : 0.03422834547294996 }
{ "_id" : ObjectId("611f3da185009a81ad38e74a"), "state" : "Washington", "city" :
  "Seattle", "name" : "Thai Palace", "rating" : 4.8, "location" : { "type" : "Point",
  "coordinates" : [ -122.3264, 47.6009 ] }, "DistanceKilometers" : 2.5009390081704277 }
{ "_id" : ObjectId("611f3dae85009a81ad38e74c"), "state" : "Washington", "city" :
  "Seattle", "name" : "Curry House", "rating" : 4.8, "location" : { "type" : "Point",
  "coordinates" : [ -121.4517, 47.6229 ] }, "DistanceKilometers" : 67.52845344856914 }
```

若要限制查詢中的結果數目，請使用limit或num選項。

limit:

```
db.usarestaurants.aggregate([
  {
    "$geoNear":{
      "near":{
        "type":"Point",
        "coordinates":[
          -122.3516,
          47.6156
        ]
      },
      "spherical":true,
      "distanceField":"DistanceKilometers",
      "distanceMultiplier":0.001,
      "limit": 10
    }
  }
])
```

num:

```
db.usarestaurants.aggregate([
  {
    "$geoNear":{
      "near":{
        "type":"Point",
        "coordinates":[
          -122.3516,
          47.6156
        ]
      },
      "spherical":true,
      "distanceField":"DistanceKilometers",
      "distanceMultiplier":0.001,
      "num": 10
    }
  }
])
```

Note

`$geoNearstage` 支援 `limit` 和 `num` 項，以指定要傳回的文件數目上限。`$geoNear` 如果未指定 `limit` 或 `num` 選項，依預設，最多會傳回 100 個文件。如果存在且值小於 100，則 `$limit` 階段值會覆寫此值。

範例 2

在此示例中，您將學習如何找到特定地址（點）2 公里內的所有餐廳（點）。要執行此類查詢，您可以 `$nearSphere` 在 GeoJSON 點的最小 `$minDistance` 和最大值 `$maxDistance` 內使用

```
db.usarestaurants.find({
  "location":{
    "$nearSphere":{
      "$geometry":{
        "type":"Point",
        "coordinates":[
          -122.3516,
          47.6156
        ]
      },
      "$minDistance":1,
    }
  }
})
```

```
        "$maxDistance":2000
    }
}
},
{
    "name":1
})
```

上面的命令將返回距指定點最大距離 2 公里的餐廳。這個命令的輸出看起來像這樣

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "name" : "Noodle House" }
```

限制

Amazon DocumentDB 不支援多邊形、`LineString MultiPoint`、`MultiPolygon`和的查詢或索引。
`MultiLineString GeometryCollection`

部分索引

部分索引會索引集合中符合指定篩選條件的文件。部分索引功能在 Amazon DocumentDB 5.0 執行個體型叢集中受支援。

主題

- [創建一個部分索引](#)
- [支援的運算子](#)
- [使用部分索引進行查詢](#)
- [部分索引功能](#)
- [部分索引限制](#)

創建一個部分索引

若要建立部分索引，請搭配`partialFilterExpression`選項使用該`createIndex()`方法。例如，下列作業會在 `order` 集合中建立唯一的複合索引，以索引具有`OrderID`且`isDelivered`欄位為 `true` 的文件：

```
db.orders.createIndex(
  {"category": 1, "CustomerId": 1, "OrderId": 1},
  {"unique": true, "partialFilterExpression":
```

```

    {"$and": [
      {"OrderId": {"$exists": true}},
      {"isDelivered": {"$eq": false}}
    ]}
  }
)

```

支援的運算子

- \$eq
- \$exists
- \$ 和 (僅在頂級)
- \$gt/\$gt/\$lt/\$lte (索引掃描僅在查詢中預測的篩選器完全符合部分篩選器運算式時才會使用) (請參閱限制)

使用部分索引進行查詢

使用部分索引可以使用以下查詢模式：

- 查詢述詞完全符合部分索引篩選器運算式：

```

db.orders.find({"$and": [
  {"OrderId": {"$exists": true}},
  {"isDelivered": {"$eq": false}}
]}).explain()

```

- 查詢篩選器的預期結果是部分篩選器的邏輯子集：

```

db.orders.find({"$and": [
  {"OrderId": {"$exists": true}},
  {"isDelivered": {"$eq": false}},
  {"OrderAmount": {"$eq": "5"}}
]}).explain()

```

- 查詢的子謂詞可以與其他索引一起使用：

```

db.orders.createIndex({"anotherIndex":1})
db.orders.find({ "$or": [
  {"$and": [

```



```
        {"orderId": {"$exists": true}},
        {"isDelivered": {"$eq": false}}
    ]},
    {"anotherIndex": {"$eq": 5}}
]
}).explain()
```

Note

如果有效率的話，查詢規劃工具可以選擇使用收集掃描而非索引掃描。對於非常小的集合或將返回集合的很大一部分的查詢，通常會出現這種情況。

部分索引功能

列出部分索引

`partialFilterExpression` 使用 `getIndex` 操作列出部分索引。例如，在中發出的 `getIndex` 作業會列出包含索引鍵、名稱和部分檔案解壓式欄位的部分索引：

```
db.orders.getIndexes()
```

此範例會傳回下列輸出：

```
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "ecommerceApp.orders"
  },
  {
    "v" : 4,
    "unique" : true,
    "key" : {
      "category" : 1,
      "" : 1,
      "CustomerId" : 1,

```

```

    "OrderId" : 1
  },
  "name" : "category_1_CustID_1_OrderId_1",
  "ns" : "ecommerceApp.orders",
  "partialFilterExpression" : {
    "$and" : [
      {"OrderId": {"$exists": true}},
      {"isDelivered": {"$eq": false}}
    ]
  }
}
]

```

同一鍵上的多個部分過濾器表達式:order

可以為相同的欄位組合 (key: order) 建立不同的部分索引。這些索引必須有不同的名稱。

```

db.orders.createIndex(
  {"OrderId":1},
  {
    name:"firstPartialIndex",
    partialFilterExpression:{"OrderId":{"$exists": true}}
  }
)

```

```

db.orders.createIndex(
  {"OrderId":1},
  {
    name:"secondPartialIndex",
    partialFilterExpression:{"OrderId":{"$gt": 1000}}
  }
)

```

運行getIndexes操作列出集合中的所有索引：

```
db.orders.getIndexes()
```

這些範例會傳回下列輸出：

```

[
  {

```

```
"v" : 4,
"key" : {
  "_id" : 1
},
"name" : "_id_",
"ns" : "ecommerceApp.orders"
},
{
  "v" : 4,
  "key" : {
    "OrderId" : 1
  },
  "name" : "firstPartialIndex",
  "ns" : "ecommerceApp.orders",
  "partialFilterExpression" : {"OrderId":{"$exists": true}}
},
{
  "v" : 4,
  "key" : {
    "OrderId" : 1
  },
  "name" : "secondPartialIndex",
  "ns" : "ecommerceApp.orders",
  "partialFilterExpression" : {"OrderId":{"$gt": 1000}}
}
]
```

Important

索引名稱必須不同，並且只能按名稱刪除。

具有部分屬性和 TTL 屬性的索引

您也可以在建立索引期間指定和`expireAfterSeconds`選項，以建立具有部分屬性`partialFilterExpression`和 TTL 屬性的索引。這使您可以更好地控制哪些文檔現在從集合中刪除。

例如，您可能有一個 TTL 索引，用於識別在特定時間段後要刪除的文件。您現在可以針對何時使用部分索引選項來刪除文件提供額外條件：

```
db.orders.createIndex(
```

```
{ "OrderTimestamp": 1 },
{
  expireAfterSeconds: 3600 ,
  partialFilterExpression: { "isDelivered": { $eq: true } }
}
)
```

此範例會傳回下列輸出：

```
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1,
  "operationTime" : Timestamp(1234567890, 1)
}
```

運行getIndexes操作以列出集合中存在的索引：

```
db.orders.getIndexes()
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "test.orders"
  }
]
```

此範例會傳回下列輸出：

```
[
  {
    "v": 4,
    "key": {
      "_id": 1
    },
    "name": "_id_",
    "ns": "ecommerceApp.orders"
  },
  {
```

```

    "v": 4,
    "key": {
      "OrderTimestamp": 1
    },
    "name": "OrderTimestamp_1",
    "ns": "ecommerceApp.orders",
    "partialFilterExpression": {
      "isDelivered": {
        "$eq": true
      }
    },
    "expireAfterSeconds": 3600
  }
]

```

部分索引限制

下列限制適用於部分索引功能：

- Amazon DocumentDB 中的不等式查詢只會在查詢篩選器述詞完全符合 `partialFilterExpression` 和是相同的資料類型時，才會使用部分索引。

Note

對於上述情況，甚至 `$hint` 不能用於強制 IXSCAN。

在下列範例中，只會 `partialFilterExpression` 套用至 `field1` 但不適用於 `field2`：

```

db.orders.createIndex(
  {"OrderAmount": 1},
  {"partialFilterExpression": { OrderAmount : {"$gt" : 5}}}
)

db.orders.find({OrderAmount : {"$gt" : 5}}) // Will use partial index
db.orders.find({OrderAmount : {"$gt" : 6}}) // Will not use partial index
db.orders.find({OrderAmount : {"$gt" : Decimal128(5.00)}}) // Will not use partial
index

```

- 不支援 `partialFilterExpression` 帶有陣列運算子的 A。下列作業會產生錯誤：

```

db.orders.createIndex(

```

```
    {"CustomerId":1},  
    {'partialFilterExpression': {'OrderId': {'$eq': [1000, 1001, 1002]}}}  
  )
```

- 欄位中不支援下列運算 `partialFilterExpression` 子：
 - `$all`(陣列運算子)
 - `$mod`(陣列運算子)
 - `$or`
 - `$xor`
 - `$not`
 - `$nor`
- 篩選運算式和篩選器的資料類型應該是相同的。

使用 Amazon DocumentDB 執行文本搜索

Amazon DocumentDB 的原生全文檢索搜尋功能可讓您使用特殊用途的文字索引，在大型文字資料集上執行文字搜尋。本節說明文字索引功能的功能，並提供如何在 Amazon DocumentDB 中建立和使用文字索引的步驟。也會列出文字搜尋限制。

主題

- [支援的功能](#)
- [使用 Amazon DocumentDB 文本索引](#)
- [與 MongoDB 的差異](#)
- [最佳做法和指引](#)
- [限制](#)

支援的功能

Amazon DocumentDB 文字搜尋支援下列與 MongoDB API 相容的功能：

- 在單一欄位上建立文字索引。
- 建立包含多個文字欄位的複合文字索引。
- 執行單字或多字搜尋。
- 使用權重控制搜尋結果。

- 按分數對搜索結果進行排序。
- 在彙總管線中使用文字索引。
- 搜索確切的短語。

使用 Amazon DocumentDB 文本索引

要在包含字符串數據的字段上創建一個文本索引，指定字符串「text」，如下所示：

單字段索引：

```
db.test.createIndex({"comments": "text"})
```

此索引支援指定集合中「Comments」字串欄位中的文字搜尋查詢。

在多個字符串字段上創建複合文本索引：

```
db.test.createIndex({"comments": "text", "title":"text"})
```

該索引支持指定集合中的「註釋」和「標題」字符串字段中的文本搜索查詢。建立複合文字索引時，您最多可以指定 30 個欄位。一旦創建，您的文本搜索查詢將查詢所有索引字段。

Note

每個集合只允許一個文字索引。

列出 Amazon DocumentDB 集合上的文本索引

您可以使用您的 `getIndexes()` 的集合來識別和描述索引，包括文本索引，如下面的例子所示：

```
rs0:PRIMARY> db.test.getIndexes()
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
```

```
    "ns" : "test.test"
  },
  {
    "v" : 1,
    "key" : {
      "_fts" : "text",
      "_ftsx" : 1
    },
    "name" : "contents_text",
    "ns" : "test.test",
    "default_language" : "english",
    "weights" : {
      "comments" : 1
    },
    "textIndexVersion" : 1
  }
]
```

一旦你創建了一個索引，開始插入數據到你的 Amazon DocumentDB 集合。

```
db.test.insertMany([{"_id": 1, "star_rating": 4, "comments": "apple is red"},
                    {"_id": 2, "star_rating": 5, "comments": "pie is delicious"},
                    {"_id": 3, "star_rating": 3, "comments": "apples, oranges - healthy fruit"},
                    {"_id": 4, "star_rating": 2, "comments": "bake the apple pie in the oven"},
                    {"_id": 5, "star_rating": 5, "comments": "interesting couch"},
                    {"_id": 6, "star_rating": 5, "comments": "interested in couch for sale, year 2022"}])
```

執行文字搜尋查詢

執行單字文字搜尋查詢

您將需要使用 `$text` 和 `$search` 運算符來執行文本搜索。下列範例會傳回文字索引欄位包含字串「apple」或「apple」字串的所有文件 (例如「apple」)：

```
db.test.find({$text: {$search: "apple"}})
```

輸出：

這個命令的輸出看起來像這樣：


```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

執行多字文字搜尋

您也可以在 Amazon 文件資料上執行多字文字搜尋。下面的命令返回包含「蘋果」或「pie」的文本索引字段的文檔：

```
db.test.find({$text: {$search: "apple pie"}})
```

輸出：

這個命令的輸出看起來像這樣：

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 2, "star_rating" : 5, "comments" : "pie is delicious" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

執行多字詞語文字搜尋

對於多字詞語搜尋，請使用以下範例：

```
db.test.find({$text: {$search: "\"apple pie\""}})
```

輸出：

上面的命令返回文本索引字段包含確切短語「apple pie」的文檔。這個命令的輸出看起來像這樣：

```
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

使用篩選器執行文字搜尋

您也可以將文字搜尋與其他查詢運算子結合使用，以根據其他條件篩選結果：

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "interest"}}]})
```

輸出：

上面的命令返回包含任何形式的「興趣」和「star_評級」等於 5 的文本索引字段的文檔。這個命令的輸出看起來像這樣：

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
{ "_id" : 6, "star_rating" : 5, "comments" : "interested in couch for sale, year
2022" }
```

限制文字搜尋中傳回的文件數

您可以使用下列方式選擇限制傳回的文件數量limit：

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "couch"}}]}).limit(1)
```

輸出：

上面的命令返回一個滿足過濾器的結果：

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
```

依文字分數排序結果

下列範例會依文字分數排序文字搜尋結果：

```
db.test.find({$text: {$search: "apple"}}, {score: {$meta: "textScore"}}).sort({score:
{$meta: "textScore"}})
```

輸出：

上面的命令返回一個文本索引字段包含「蘋果」，或「蘋果」在其他格式（如「蘋果」）的文檔，並根據文檔與搜索詞的相關性對結果進行排序。這個命令的輸出看起來像這樣：

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red", "score" :
0.6079270860936958 }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit",
"score" : 0.6079270860936958 }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven",
"score" : 0.6079270860936958 }
```

\$text和\$search也支援aggregate、count、findAndModifyupdate、和delete指令。

聚合運算子

使用聚合管線 \$match

```
db.test.aggregate(  
  [ { $match: { $text: { $search: "apple pie" } } } ]  
)
```

輸出：

上面的命令返回以下結果：

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }  
{ "_id" : 3, "star_rating" : 3, "comments" : "apple - a healthy fruit" }  
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }  
{ "_id" : 2, "star_rating" : 5, "comments" : "pie is delicious" }
```

其他聚合運算符的組合

```
db.test.aggregate(  
  [  
    { $match: { $text: { $search: "apple pie" } } },  
    { $sort: { score: { $meta: "textScore" } } },  
    { $project: { score: { $meta: "textScore" } } }  
  ]  
)
```

輸出：

上面的命令返回以下結果：

```
{ "_id" : 4, "score" : 0.6079270860936958 }  
{ "_id" : 1, "score" : 0.3039635430468479 }  
{ "_id" : 2, "score" : 0.3039635430468479 }  
{ "_id" : 3, "score" : 0.3039635430468479 }
```

建立文字索引時指定多個欄位

複合文字索引中最多可以指定三個欄位的線寬。指定給文字索引中欄位的預設寬度為一 (1)。重量是一個可選的參數，並且必須在 1 到 100000 的範圍內。

```
db.test.createIndex(  
  {  
    "firstname": "text",  
    "lastname": "text",  
    ...  
  },  
  {  
    weights: {  
      "firstname": 5,  
      "lastname":10,  
      ...  
    },  
    name: "name_text_index"  
  }  
)
```

與 MongoDB 的差異

Amazon DocumentDB 的文本索引功能使用反轉索引和術語頻率算法。根據預設，文字索引是稀疏的。由於在解析邏輯，標記分隔符，和其他的差異，相同的結果集 MongoDB 可能不會被返回相同的數據集或查詢形狀。

Amazon DocumentDB 文本索引和 MongoDB 之間存在以下附加差異：

- 不支援使用非文字索引的複合索引。
- Amazon DocumentDB 文本索引不區分大小寫，並且不區分變音符號。
- 文字索引僅支援英文。
- 不支援陣列 (或多鍵) 欄位的文字索引。例如，在「a」上創建文本索引，文檔 {「a」:[「蘋果」，「pie」]} 將失敗。
- 不支援萬用字元文字索引。
- 不支援唯一文字索引。
- 不支援排除字詞。

最佳做法和指引

- 為了在涉及按文字分數排序的文字搜尋查詢上獲得最佳效能，我們建議您在載入資料之前先建立文字索引。
- 文字索引需要額外的儲存空間，才能最佳化索引資料的內部副本。這有額外的成本影響。

限制

文字搜尋在 Amazon DocumentDB 中有下列限制：

- 只有 Amazon DocumentDB 5.0 執行個體型叢集才支援文字搜尋。

Amazon DocumentDB 疑難排解

以下各節提供有關如何疑難排解使用 Amazon DocumentDB (與 MongoDB 相容性) 時可能遇到的問題的相關資訊。

主題

- [連線問題](#)
- [索引建立](#)
- [效能與資源使用率](#)

連線問題

連線時遇到問題嗎？以下為一些常發生的情況及其解決方法。

主題

- [無法連接到 Amazon DocumentDB 端點](#)
- [測試 Amazon DocumentDB 執行個體的連線](#)
- [連線到無效的端點](#)

無法連接到 Amazon DocumentDB 端點

當您嘗試連線到 Amazon DocumentDB 時，下列是您可能會收到的最常見錯誤訊息之一。

```
connecting to: mongodb://docdb-2018-11-08-21-47-27.cluster-ccuszb3pn5e.us-east-1.docdb.amazonaws.com:27017/
2018-11-14T14:33:46.451-0800 W NETWORK [thread1] Failed to connect to
172.31.91.193:27017 after 5000ms milliseconds, giving up.
2018-11-14T14:33:46.452-0800 E QUERY [thread1] Error: couldn't connect to server
docdb-2018-11-08-21-47-27.cluster-ccuszb3pn5e.us-east-1.docdb.amazonaws.com:27017,
connection attempt failed :
connect@src/mongo/shell/mongo.js:237:13
@(connect):1:6
exception: connect failed
```

此錯誤訊息通常意味著您的用戶端 (本範例中的 mongo 殼層) 無法存取 Amazon DocumentDB 端點。這可能有幾個原因：

主題

- [從公用端點連線](#)
- [跨區域連接](#)
- [從不同的亞馬遜 VPC 連接](#)
- [安全群組封鎖輸入連線](#)
- [Java 蒙戈驅動程序讀取首選項問題](#)

從公用端點連線

您嘗試直接從筆記型電腦或本機開發機器連線到 Amazon DocumentDB 叢集。

嘗試直接從公用端點 (例如筆記型電腦或本機開發機器) 連線到 Amazon DocumentDB 叢集將會失敗。Amazon DocumentDB 僅限虛擬私有雲端 (VPC)，目前不支援公有端點。因此，您無法從筆記型電腦或 VPC 以外的本機開發環境直接連線到 Amazon DocumentDB 叢集。

若要從亞馬遜 VPC 擬私人雲端外部連線到 Amazon 文件資料庫叢集，您可以使用安全殼層通道。如需詳細資訊，請參閱[從亞馬遜 VPC 外部連接到 Amazon DocumentDB 集群](#)。此外，如果您的開發環境位於不同的 Amazon VPC 中，您也可以使用 VPC 對等互連，並從相同區域或不同區域的另一個 Amazon VPC 連接到 Amazon DocumentDB 叢集。

跨區域連接

您嘗試連線到另一個區域中的 Amazon DocumentDB 叢集。

如果您嘗試從叢集區域以外區域 (us-west-2) 區域 (us-west-2) 內 Amazon EC2 執行個體 (例如，嘗試從美國西 Amazon DocumentDB (奧勒岡) 區域 (us-west-2) 連線到美國東部 (維吉尼亞北部) 區域 (us-east-1) 內的叢集。

若要驗證您的 Amazon DocumentDB 叢集的區域，請執行以下命令。區域在端點中。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].Endpoint'
```

此操作的輸出將會如下所示。

```
[  
  "sample-cluster.node.us-east-1.docdb.amazonaws.com"
```

```
]
```

為了驗證您的 EC2 執行個體的區域，請執行以下命令。

```
aws ec2 describe-instances \  
  --query 'Reservations[*].Instances[*].Placement.AvailabilityZone'
```

此操作的輸出將會如下所示。

```
[  
  [  
    "us-east-1a"  
  ]  
]
```

從不同的亞馬遜 VPC 連接

您嘗試從虛擬私人雲端連線到 Amazon DocumentDB 叢集，該虛擬私人雲端與叢集部署到的 Amazon VPC 不同。

如果您的 Amazon DocumentDB 叢集和 Amazon EC2 執行個體位於相同的執行個體AWS 區域，但不在同一個 Amazon VPC 中，則除非在兩個 Amazon VPC 之間啟用了 VPC 對等互連，否則您無法直接連線到 Amazon DocumentDB 叢集。

若要驗證您的 Amazon DocumentDB 執行個體的 Amazon VPC，請執行以下命令。

```
aws docdb describe-db-instances \  
  --db-instance-identifier sample-instance \  
  --query 'DBInstances[*].DBSubnetGroup.VpcId'
```

若要驗證您的 Amazon EC2 執行個體的 Amazon VPC，請執行以下命令。

```
aws ec2 describe-instances \  
  --query 'Reservations[*].Instances[*].VpcId'
```

安全群組封鎖輸入連線

您嘗試連線到 Amazon DocumentDB 叢集，且叢集的安全群組不允許在叢集的連接埠上進行輸入連線 (預設連接埠：27017)。

假設您的 Amazon DocumentDB 叢集和 Amazon EC2 執行個體都位於同一個區域和亞馬遜 VPC，並且使用相同的 Amazon VPC 安全群組。如果您無法連線到 Amazon DocumentDB 叢集，可能的原因是叢集的安全群組 (也就是防火牆) 不允許在您為 Amazon DocumentDB 叢集選擇的連接埠上輸入連線 (預設連接埠為 27017)。

若要驗證您的 Amazon DocumentDB 叢集的連線，請執行以下命令。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[DBClusterIdentifier,Port]'
```

若要為您的叢集取得您的 Amazon DocumentDB 安全群組，請執行以下命令。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[VpcSecurityGroups[*],VpcSecurityGroupId]'
```

若要檢查安全群組規則的傳入規則，請參閱 Amazon EC2 說明文件中的以下主題：

- [授權 Linux 執行個體的輸入流量](#)
- [授權 Windows 執行個體的輸入流量](#)

Java 蒙戈驅動程序讀取首選項問題

不會遵守用戶端讀取喜好設定，且有些用戶端在容錯移轉後無法寫入 Amazon DocumentDB，除非他們重新開機。

這個問題，首先在 Java 蒙戈驅動程式 3.7.x 中發現，當用戶端建立連接到 Amazon DocumentDB 使用 MongoClientSettings 和，特別是，鏈接方法時發生。applyToClusterSettings 您可以使用幾種不同的方法來定義 MongoClient 叢集設定 hosts()，例如 requiredReplicaSetName()、和 mode()。

當用戶端在 hosts() 方法中僅指定一台主機時，模式會設定為 ClusterConnectionMode.SINGLE 而非 ClusterConnectionMode.MULTIPLE 這會導致用戶端忽略讀取偏好設定，而只連線至 hosts() 設定的伺服器。因此，即使客戶端設置如下所示初始化，所有讀取仍然會轉到主讀取而不是次要讀取。

```
final ServerAddress serverAddress0 = new ServerAddress("cluster-endpoint", 27317));  
final MongoCredential credential = MongoCredential.createCredential("xxx",
```

```
        "admin", "xxxx".toCharArray());
final MongoClientSettings settings = MongoClientSettings.builder()
    .credential(credential)
    .readPreference(ReadPreference.secondaryPreferred())
    .retryWrites(false)
    .applyToSslSettings(builder -> builder
        .enabled(false))
    .applyToClusterSettings(builder -> builder.hosts(
        Arrays.asList(serverAddress0
            ))
        .requiredReplicaSetName("rs0"))
    .build();
MongoClient mongoClient = MongoClients.create(settings);
```

容錯移轉案

使用上述用戶端連線設定時，如果叢集寫入器端點有容錯移轉和延遲 DNS 記錄更新，用戶端仍會嘗試寫入舊寫入器 (現在容錯移轉後的讀取器)。這會導致服務器端錯誤 (不是 master)，Java 驅動程序未正確處理 (這仍在調查中)。因此，例如，在重新啟動應用程式伺服器之前，用戶端可能會處於不良狀態。

有兩種解決方法：

- 透過連接字串連線至 Amazon DocumentDB 的用戶端將不會出現此問題，因為在設定讀取偏好設定 MULTIPLE 時 ClusterConnectionMode 會設為。

```
MongoClientURI mongoClientURI = new MongoClientURI("mongodb://usr:pass:cluster-  
endpoint:27317/test?ssl=false&replicaSet=rs0&readPreference=secondaryPreferred");  
MongoClient mongoClient = MongoClients.create(mongoClientURI.getURI());
```

或 MongoClientSettings 者使用構建器與 applyConnectionString 方法。

```
final MongoClientSettings settings = MongoClientSettings.builder()
    .credential(credential)
    .applyConnectionString(new ConnectionString("usr:pass:cluster-endpoint:27317/  
test?ssl=false&replicaSet=rs0&readPreference=secondaryPreferred"))
    .retryWrites(false)
    .applyToSslSettings(builder # builder
        .enabled(false))
    .build();
MongoClient mongoClient = MongoClients.create(settings);
```

- 明確設定ClusterConnectionMode為MULTIPLE。這只有在使用applyToClusterSettings和時才需要hosts().size() == 1。

```
final ServerAddress serverAddress0 = new ServerAddress("cluster-endpoint", 27317));
final MongoCredential credential = MongoCredential.createCredential("xxx", "admin",
    "xxxx".toCharArray());
final MongoClientSettings settings = MongoClientSettings.builder()
    .credential(credential)
    .readPreference(ReadPreference.secondaryPreferred())
    .retryWrites(false)
    .applyToSslSettings(builder # builder
        .enabled(false))
    .applyToClusterSettings(builder # builder
        .hosts(Arrays.asList(serverAddress0))
        .requiredReplicaSetName("rs0"))
    .mode(ClusterConnectionMode.MULTIPLE))
    .build();
MongoClient mongoClient = MongoClient.create(settings);
```

測試 Amazon DocumentDB 執行個體的連線

您可以使用通用的 Linux 或 Windows 工具來測試您的叢集連線。

從 Linux 或 Unix 終端機測試連線，方法是輸入下列命令 (以端點取代 cluster-endpoint，以及以執行個體的連接埠取代 port)：

```
nc -zv cluster-endpoint port
```

以下是範例操作和傳回值的範例：

```
nc -zv docdbTest.d4c7nm7stsf0.us-west-2.docdb.amazonaws.com 27017

Connection to docdbTest.d4c7nm7stsf0.us-west-2.docdb.amazonaws.com 27017 port [tcp/*]
succeeded!
```

連線到無效的端點

連線到 Amazon DocumentDB 叢集並使用無效的叢集端點時，會出現類似下列內容的錯誤。

```
mongo --ssl \
```

```
--host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 \  
--sslCAFile global-bundle.pem \  
--username <user-name> \  
--password <password>
```

輸出如下：

```
MongoDB shell version v3.6  
connecting to: mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/  
2018-11-14T17:21:18.516-0800 I NETWORK [thread1] getaddrinfo("sample-cluster.node.us-  
east-1.docdb.amazonaws.com") failed:  
nodename nor servname provided, or not known 2018-11-14T17:21:18.537-0800 E QUERY  
[thread1] Error: couldn't initialize  
connection to host sample-cluster.node.us-east-1.docdb.amazonaws.com, address is  
invalid :  
connect@src/mongo/shell/mongo.js:237:13@(connect):1:6  
exception: connect failed
```

若要取得有效的叢集端點，請執行下列命令：

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[Endpoint,Port]'
```

若要取得有效的執行個體端點，請執行下列命令：

```
aws docdb describe-db-instances \  
  --db-instance-identifier sample-instance \  
  --query 'DBInstances[*].[Endpoint.Address,Endpoint.Port]'
```

如需詳細資訊，請參閱[了解亞馬遜文檔數據庫端點](#)。

索引建立

下列主題討論，如果您的索引或背景索引建置失敗，怎麼辦。

主題

- [索引組建失敗](#)
- [背景索引建置延遲問題和失敗](#)

索引組建失敗

Amazon DocumentDB 可利用執行個體上的本機儲存，這是索引建立程序的一部分。您可以使用 `FreeLocalStorageCloudWatch` 指標 (CloudWatch -> Metrics -> DocDB -> Instance Metrics) 來監視此磁碟使用狀況。當索引建置使用所有本機磁碟並失敗時，您將會收到錯誤。將資料遷移到 Amazon DocumentDB 時，我們建議您先建立索引，然後再插入資料。如需有關移轉策略和建立索引的詳細資訊，請參閱 Amazon DocumentDB 文件和部落格 [遷移到 Amazon DocumentDB](#) 中的：使用離線方法 [從 MongoDB 遷移到 Amazon DocumentDB](#)。

在現有叢集上建立索引時，如果索引建置花費的時間超過預期或失敗，建議您擴展執行個體以建立索引，然後在建立索引之後縮減。Amazon DocumentDB 可讓您在幾分鐘內使用 AWS Management Console 或快速擴展執行個體大小。AWS CLI 如需詳細資訊，請參閱 [管理執行個體類別](#)。透過每秒執行個體定價，您只需支付使用的資源，以秒計費。

背景索引建置延遲問題和失敗

Amazon DocumentDB 中的背景索引建置要等到在索引建立起始之前啟動的主執行個體上的所有查詢完成執行才會啟動。如果有長時間執行的查詢，背景索引建置會封鎖直到查詢完成，因此可能需要比預期更長的時間才能完成。即使集合是空的也是如此。

前景索引組建不會表現出相同的封鎖行為。相反地，前景索引建置會在集合上採用獨佔鎖定，直到索引建置完成為止。因此，為了在空集合上創建索引並避免阻塞任何長時間運行的查詢，我們建議使用前景索引構建。

Note

Amazon DocumentDB 只允許在任何給定時間在集合上建立一個背景索引。如果在背景索引建置期間 DDL (資料定義語言) 操作 (如 `createIndex()` 或 `dropIndex()`) 發生於同一個集合，則背景索引建置會失敗。

效能與資源使用率

本節提供 Amazon DocumentDB 部署中常見診斷問題的問題和解決方案。這些範例會提供使用 mongo 殼層，而且範圍限定在個別執行個體的示範。若要尋找執行個體端點，請參閱 [了解亞馬遜文檔數據庫端點](#)。

主題

- [如何確定通過 Mongo API 在我的集合上執行的插入，更新和刪除操作的數量？](#)
- [如何分析快取效能？](#)
- [如何尋找並終止長時間執行或封鎖的查詢？](#)
- [如何查看查詢計劃並優化查詢？](#)
- [如何在彈性集群中查看查詢計劃？](#)
- [如何列出執行個體上所有正在執行的操作？](#)
- [我如何知道查詢何時正在進行？](#)
- [如何確定系統突然運行緩慢的原因？](#)
- [如何判斷一或多個叢集執行個體上 CPU 使用率過高的原因？](#)
- [如何確定實例上的打開游標？](#)
- [如何確定當前的 Amazon DocumentDB 引擎版本？](#)
- [如何分析索引使用情況並識別未使用的索引？](#)
- [如何識別遺失的索引？](#)
- [有用的查詢摘要](#)

如何確定通過 Mongo API 在我的集合上執行的插入，更新和刪除操作的數量？

若要檢視對特定集合執行的插入、更新和刪除作業數目，請在該集合上執行下列命令：

```
db.collection.stats()
```

此命令的輸出描述了其opCounters字段下的以下內容：

- numDocsIns-插入此集合中的文檔數量。這包括使用insert和insertMany指令插入的文件，以及upsert 插入的文件。
- numDocsUpd-此集合中的文檔更新數量。這包括使用update和findAndModify指令更新的文件。
- numDocsDel-從此集合中刪除的文檔數量。這包括使用deleteOne、deleteMany、remove和findAndModify指令刪除的文件。
- LaStreset-這些計數器上次重設的時間。啟動/停止叢集或擴展/縮減執行個體時，會重設此命令提供的統計資料。

從運行db.collection.stats()的一個例子輸出如下所示。

```
{
  "ns" : "db.test",
  "count" : ...,
  "size" : ...,
  "avgObjSize" : ...,
  "storageSize" : ...,
  "capped" : false,
  "nindexes" : ...,
  "totalIndexSize" : ...,
  "indexSizes" : {
    "_id_" : ...,
    "x_1" : ...
  },
  "collScans" : ...,
  "idxScans" : ...,
  "opCounter" : {
    "numDocsIns" : ...,
    "numDocsUpd" : ...,
    "numDocsDel" : ...
  },
  "cacheStats" : {
    "collBlksHit" : ...,
    "collBlksRead" : ..,
    "collHitRatio" : ...,
    "idxBlksHit" : ...,
    "idxBlksRead" : ...,
    "idxHitRatio" : ...
  },
  "lastReset" : "2022-09-02 19:41:40.471473+00",
  "ok" : 1,
  "operationTime" : Timestamp(1662159707, 1)
}
```

通過 Mongo API 查看集合特定計數器插入，更新和刪除操作時，應使用此統計命令。另一種檢視集合特定作業計數器的方法是啟用 DML 稽核。您可以在中檢視一分鐘時間間隔內對所有集合的插入、更新和刪除作業數目 [使用監控亞馬遜文檔數據庫 CloudWatch](#)。

如何分析快取效能？

分析快取效能可以深入瞭解資料擷取的效率和系統效能，並以從磁碟讀取的資料量與快取為基礎。我們提供有關緩存命中數（從緩存中讀取的數據）和緩存未命中（未在緩存中找到並從磁碟讀取數據）的緩

存統計信息，以便深入了解緩存性能。特定集合的快取統計資料可以透過在該集合上執行下列命令來找到：

```
db.collection.stats()
```

此命令輸出中cacheStats欄位中的值會提供集合的快取統計資料，以及在集合上建立之索引的總快取統計資料。這些統計數據列舉如下：

- **collBlksHit**-執行此集合作業期間，從快取讀取的區塊數目。
- **collBlksRead**-在此集合作業期間，從磁碟讀取的區塊數目 (快取遺漏)。
- **collHitRatio**-此集合 ($100 * [\text{collBlksHit} / (\text{collBlksHit} + \text{collBlksRead})]$) 的快取命中率。
- **idxBlksHit**-針對此集合上建立的任何索引，從快取讀取的區塊數目。
- **idxBlksRead**-針對此集合上建立的任何索引，從磁碟讀取的區塊數目 (快取遺漏)。
- **idxHitRatio**-在此集合 () 上建立之索引的快取命中 $100 * [\text{idxBlksHit} / (\text{idxBlksHit} + \text{idxBlksRead})]$ 率。
- **lastReset**-上次重設這些統計資料的時間。啟動/停止叢集或向上/縮減執行個體時，會重設提供的統計資料。db.collection.stats()

您也可以使用指indexStats令找到每個索引的idxBlksHit和idxBlksRead欄位劃分。索引特定的快取統計資料可以透過執行下列命令找到：

```
db.collection.aggregate([{$indexStats: {}}]).pretty()
```

對於每個索引，您可以在cacheStats欄位下找到下列快取統計資料：

- **blksHit**-從快取中讀取此索引的區塊數目。
- **blksRead**-從磁碟讀取此索引的區塊數目。
- **blksHitRatio**-快取命中率四捨五入至四位小數，計算方式為 $100 * [\text{blksHit} / (\text{blksHit} + \text{blksRead})]$ 。

如何尋找並終止長時間執行或封鎖的查詢？

使用者查詢執行緩慢的原因，可能是查詢計畫不佳，或因為資源爭用而遭封鎖。

若要尋找因查詢計畫不佳而造成長時間執行查詢的效率變慢，或因為資源爭用而遭封鎖的查詢，請先使用 `currentOp` 命令。您可以篩選命令，以利減縮要終止的相關查詢清單。您必須具備與長時間執行查詢相關聯的 `opid`，才能夠終止查詢。

以下查詢會使用 `currentOp` 命令來列出所有遭到封鎖或執行超過 10 秒的查詢。

```
db.adminCommand({
  aggregate: 1,
  pipeline: [
    {$currentOp: {}},
    {$match:
      {$or: [
        {secs_running: {$gt: 10}},
        {WaitState: {$exists: true}}]}]},
    {$project: {_id:0, opid: 1, secs_running: 1}},
  ],
  cursor: {}
});
```

接著，您就可以縮減查詢，尋找並終止執行超過 10 秒的查詢 `opid`。

尋找和終止執行超過 10 秒的查詢

1. 尋找查詢的 `opid`。

```
db.adminCommand({
  aggregate: 1,
  pipeline: [
    {$currentOp: {}},
    {$match:
      {$or:
        [{secs_running: {$gt: 10}},
        {WaitState: {$exists: true}}]}]},
  ],
  cursor: {}
});
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "waitedMS" : NumberLong(0),
  "cursor" : {
    "firstBatch" : [
      {
```

```
        "opid" : 24646,  
        "secs_running" : 12  
      }  
    ],  
    "id" : NumberLong(0),  
    "ns" : "admin.$cmd"  
  },  
  "ok" : 1  
}
```

2. 使用 killOp 操作終止查詢。

```
db.adminCommand({killOp: 1, op: 24646});
```

如何查看查詢計劃並優化查詢？

查詢的執行速度如果緩慢，可能是由於查詢執行需要完整掃描集合來選擇相關的文件。有時，建立適當的索引能夠使查詢的執行速度更快。若要偵測這種情況並決定建立索引的欄位，請使用 `explain` 命令。

Note

Amazon DocumentDB 會在使用分散式容錯、自我修復儲存系統的專用資料庫引擎上模擬 MongoDB 3.6 API。因此，Amazon DocumentDB 和 MongoDB 之間的查詢計劃和輸出 `explain()` 可能會有所不同。想要控制其查詢計劃的客戶可以使用 `$hint` 運算子強制選取偏好的索引。

請在 `explain` 命令下執行您要改善的查詢，如下所示。

```
db.runCommand({explain: {<query document>}})
```

下面是範例操作。

```
db.runCommand({explain:{  
  aggregate: "sample-document",  
  pipeline: [{$match: {x: {$eq: 1}}}],  
  cursor: {batchSize: 1}}  
});
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "db.test",
    "winningPlan" : {
      "stage" : "COLLSCAN"
    }
  },
  "serverInfo" : {
    "host" : "...",
    "port" : ...,
    "version" : "..."
  },
  "ok" : 1
}
```

上述輸出表示 \$match 階段需要掃描完整的集合，並確認每個文件中的欄位 "x" 皆等於 1。如果集合中有多個文件，則集合掃描速度 (和必然的整體查詢效能) 將非常緩慢。因此，explain 命令輸出中的 "COLLSCAN" 表示可以透過建立適當的索引來提高查詢效能。

在此範例中，查詢會檢查所有文件中的欄位 "x" 是否等於 1。因此建立欄位 "x" 的索引，便可讓查詢避免完整集合掃描，並且使用索引加快相關文件傳回。

在欄位 "x" 上建立索引後，explain 輸出如下所示。

```
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "db.test",
    "winningPlan" : {
      "stage" : "IXSCAN",
      "indexName" : "x_1",
      "direction" : "forward"
    }
  },
  "serverInfo" : {
    "host" : "...",
    "port" : ...,
    "version" : "..."
  },
  "ok" : 1
}
```

```
}
```

在欄位 "x" 上建立索引將使 \$match 階段能夠使用索引掃描，以減少必須進行評估之述詞 "x = 1" 的文件數量。

對於小型集合，如果效能提升可忽略不計，Amazon DocumentDB 查詢處理器可以選擇不使用索引。

如何在彈性集群中查看查詢計劃？

若要檢查彈性叢集中的查詢計劃，請使用 explain 命令。以下是針對分片集合的 find 查詢 explain 操作的示例操作：

```
db.runCommand(  
  {  
    explain: { find: "cities", filter: {"name": "Seoul"}}  
  }  
)
```

Note

Amazon DocumentDB 模擬 MongoDB 在專門建立的數據庫引擎上。因此，Amazon DocumentDB 和 MongoDB 之間的查詢計劃和輸出 explain() 可能會有所不同。您可以使用 \$hint 運算子來強制選取偏好索引來控制查詢計劃。

此操作的輸出可能如下所示 (JSON 格式)：

```
{  
  "queryPlanner" : {  
    "elasticPlannerVersion" : 1,  
    "winningPlan" : {  
      "stage" : "SINGLE_SHARD",  
      "shards" : [  
        {  
          "plannerVersion" : 1,  
          "namespace" : "population.cities",  
          "winningPlan" : {  
            "stage" : "SHARD_MERGE",  
            "shards" : [  
              {  
                "shardName" : "f2cf5cfd-fe9c-40ca-b4e5-298ca0d11111",
```

```
    "plannerVersion" : 1,
    "namespace" : "population.cities",
    "winningPlan" : {
      "stage" : "PARTITION_MERGE",
      "inputStages" : [
        {
          "stage" : "COLLSCAN",
          "partitionCount" : 21
        }
      ]
    }
  },
  {
    "shardName" : "8f3f80e2-f96c-446e-8e9d-aab8c7f22222",
    "plannerVersion" : 1,
    "namespace" : "population.cities",
    "winningPlan" : {
      "stage" : "PARTITION_MERGE",
      "inputStages" : [
        {
          "stage" : "COLLSCAN",
          "partitionCount" : 21
        }
      ]
    }
  },
  {
    "shardName" : "32c5a06f-1b2b-4af1-8849-d7c4a0333333",
    "plannerVersion" : 1,
    "namespace" : "population.cities",
    "winningPlan" : {
      "stage" : "PARTITION_MERGE",
      "inputStages" : [
        {
          "stage" : "COLLSCAN",
          "partitionCount" : 22
        }
      ]
    }
  }
],
  "shardName" : "32c5a06f-1b2b-4af1-8849-d7c4a0f3fb58"
}
```

```
    ]
  }
},
"serverInfo" : {
  "host" : "example-4788267630.us-east-1.docdb-elastic.amazonaws.com:27017",
  "version" : "5.0.0"
},
"ok" : 1,
"operationTime" : Timestamp(1695097923, 1)
}
```

上述輸出會顯示三分片叢集上find查詢的查詢計畫。每個分片都有多個數據分區，可以具有不同的輸入階段。在此範例中，在每個碎片內的「PARTITION_MERGE」階段合併結果之前，會在所有磁碟分割上執行「COLLSCAN」(集合掃描)。然後，碎片中的結果將在「SHARD_MERGE」階段合併在一起，然後再發送回客戶端。

如何列出執行個體上所有正在執行的操作？

身為使用者或主要使用者，您通常會想要列出執行個體上執行的所有目前作業，以便進行診斷和疑難排解。(如需有關管理使用者的詳細資訊，請參閱 [Amazon DocumentDB 資料庫](#)。)

使用命mongo令介面，您可以使用下列查詢列出 Amazon DocumentDB 執行個體上所有執行中的操作。

```
db.adminCommand({currentOp: 1, $all: 1});
```

此查詢會傳回一份完整清單，內含所有使用者查詢和執行個體目前正在操作的內部系統任務。

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "inprog" : [
    {
      "desc" : "INTERNAL"
    },
    {
      "desc" : "TTLMonitor",
      "active" : false
    },
    {
      "client" : "...",
      "desc" : "Conn",
```

```
    "active" : true,
    "killPending" : false,
    "opid" : 195,
    "ns" : "admin.$cmd",
    "command" : {
      "currentOp" : 1,
      "$all" : 1
    },
    "op" : "command",
    "$db" : "admin",
    "secs_running" : 0,
    "microsecs_running" : NumberLong(68),
    "clientMetaData" : {
      "application" : {
        "name" : "MongoDB Shell"
      },
      "driver" : {
        ...
      },
      "os" : {
        ...
      }
    }
  },
  {
    "desc": "GARBAGE_COLLECTION",
    "garbageCollection": {
      "databaseName": "testdb",
      "collectionName": "testCollectionA"
    },
    "secs_running": 3,
    "microsecs_running": NumberLong(3123456)
  },
  {
    "desc": "GARBAGE_COLLECTION",
    "garbageCollection": {
      "databaseName": "testdb",
      "collectionName": "testCollectionB"
    },
    "secs_running": 4,
    "microsecs_running": NumberLong(4123456)
  }
],
"ok" : 1
```

```
}
```

以下是 "desc" 欄位的有效值：

- **INTERNAL**— 內部系統任務，如光標清理或過時的用戶清理任務。
- **TTLMonitor**— 存活時間 (TTL) 監視執行緒。其執行狀態會反映在 "active" 欄位。
- **GARBAGE_COLLECTION**— 內部記憶體回收器執行緒。
- **CONN**— 使用者查詢。
- **CURSOR**— 該操作是空間的光標，等待用戶調用「getMore」命令以獲取下一批結果。在此狀態下，游標正在消耗記憶體，但不會消耗任何計算。

前述輸出也會列出系統中執行的所有使用者查詢。每個使用者查詢會在資料庫和集合的內容中執行，這兩者的聯集稱為命名空間。"ns" 欄位會提供每個使用者查詢的命名空間。

有時您需要列出在特定命名空間中執行的所有使用者查詢。因此，之前的輸出必須經過 "ns" 欄位篩選。下面是為達到篩選輸出的範例查詢。查詢會列出資料庫 "db" 和集合 "test" (即 "db.test" 命名空間) 中目前正在執行的所有使用者查詢。

```
db.adminCommand({aggregate: 1,
  pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
    {$match: {ns: {$eq: "db.test"}}}],
  cursor: {}
});
```

身為系統的主要使用者，您可以查看所有使用者的查詢，以及所有內部系統作業。所有其他使用者都只能查看其各自查詢。

如果查詢和內部系統任務的總數超過預設批次處理游標大小，則 mongo 殼層將自動產生迭代器物件 'it' 以查看其餘結果。請持續執行命令 'it'，直到所有結果都已用盡。

我如何知道查詢何時正在進行？

使用者查詢執行緩慢的原因，可能是查詢計畫不佳，或因為資源爭用而遭封鎖。偵錯此類查詢需要進行多個步驟，而其有時需要多次執行相同的步驟。

偵錯的第一步是列出長時間執行或封鎖的所有查詢。下列查詢列出執行超過 10 秒或等待資源的所有使用者查詢。

```
db.adminCommand({aggregate: 1,
```



```

    pipeline: [{$currentOp: {}},
      {$match: {$or: [{secs_running: {$gt: 10}},
        {WaitState: {$exists: true}}]}]},
      {$project: {_id:0,
        opid: 1,
        secs_running: 1,
        WaitState: 1,
        blockedOn: 1,
        command: 1}}],
    cursor: {}
  });

```

請定期重複上述查詢，判斷查詢清單是否變更，並找出長時間執行的查詢或遭封鎖的查詢。

如果您查詢的輸出文件中包含 WaitState 欄位，則表示資源爭用是造成查詢執行速度緩慢或遭封鎖的原因。資源爭用的可能成因是 IO、內部系統任務或其他使用者查詢。

此操作的輸出將會如下所示 (JSON 格式)。

```

{
  "waitedMS" : NumberLong(0),
  "cursor" : {
    "firstBatch" : [
      {
        "opid" : 201,
        "command" : {
          "aggregate" : ...
        },
        "secs_running" : 208,
        "WaitState" : "IO"
      }
    ],
    "id" : NumberLong(0),
    "ns" : "admin.$cmd"
  },
  "ok" : 1
}

```

如果同時在相同執行個體上執行的不同集合中有多個查詢，或執行個體大小對執行查詢的資料集而言太小，則 I/O 可能成為瓶頸。如果查詢是唯讀查詢，則您可以在不同複本上分隔每個集合的查詢，有效緩解這個情況。若是跨不同集合的並行更新，或執行個體的大小對資料集而言太小，則您可以向上擴展執行個體來進行緩解。

如果資源爭用是由其他使用者查詢所引起，則輸出文件中的 "blockedOn" 欄位將包含影響此查詢之查詢的 "opid"。使用 "opid" 追蹤所有查詢的 "waitState" 和 "blockedOn" 欄位鏈結，以在鏈結的前端尋找查詢。

如果鏈結前端的任務為內部任務，則唯一的緩解措施是終止查詢，並在一段時間後重新執行該查詢。

在以下的範例輸出中，其尋找查詢在另一個任務擁有的集合鎖上遭到封鎖。

```
{
  "inprog" : [
    {
      "client" : "...",
      "desc" : "Conn",
      "active" : true,
      "killPending" : false,
      "opid" : 75,
      "ns" : "...",
      "command" : {
        "find" : "...",
        "filter" : {

        }
      },
      "op" : "query",
      "$db" : "test",
      "secs_running" : 9,
      "microsecs_running" : NumberLong(9449440),
      "threadId" : 24773,
      "clientMetaData" : {
        "application" : {
          "name" : "MongoDB Shell"
        },
        "driver" : {
          ...
        },
        "os" : {
          ...
        }
      },
      "WaitState" : "CollectionLock",
      "blockedOn" : "INTERNAL"
    },
  ]
}
```

```
        "desc" : "INTERNAL"
    },
    {
        "client" : "...",
        ...
        "command" : {
            "currentOp" : 1
        },
        ...
    }
],
"ok" : 1
}
```

如果 "WaitState" 具有值

"Latch"、"SystemLock"、"BufferLock"、"BackgroundActivity" 或 "Other"，則資源爭用來源為內部系統任務。如果情況持續很長一段時間，則唯一的緩解措施是終止該查詢，並在一段時間後重新執行該查詢。

如何確定系統突然運行緩慢的原因？

下面列出系統變慢的一些常見原因：

- 並行查詢之間的資源爭用過多
- 作用中並行查詢的數量會隨著時間增加
- 內部系統任務，例如 "GARBAGE_COLLECTION"

若要隨著時間監控系統使用量，請定期執行下列 "currentOp" 查詢，並將結果輸出到外部存放區。該查詢會計算系統中每個命名空間的查詢和操作數量。接著，您就可以分析系統使用量結果來了解系統負載，並採取適當行動。

```
db.adminCommand({aggregate: 1,
                  pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                             {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
"$WaitState"}, count: {$sum: 1}}}],
                  cursor: {}
});
```

此查詢會傳回每個命名空間中執行之所有查詢和所有內部系統任務的彙總，以及每個命名空間專屬的等待狀態數量。

此操作的輸出將會如下所示 (JSON 格式)。

```
{
  "waitedMS" : NumberLong(0),
  "cursor" : {
    "firstBatch" : [
      {
        "_id" : {
          "desc" : "Conn",
          "ns" : "db.test",
          "WaitState" : "CollectionLock"
        },
        "count" : 2
      },
      {
        "_id" : {
          "desc" : "Conn",
          "ns" : "admin.$cmd"
        },
        "count" : 1
      },
      {
        "_id" : {
          "desc" : "TTLMonitor"
        },
        "count" : 1
      }
    ],
    "id" : NumberLong(0),
    "ns" : "admin.$cmd"
  },
  "ok" : 1
}
```

在前述輸出中，命名空間 "db.test" 中有 2 個使用者查詢在集合鎖定上遭到封鎖：其中 1 個查詢位於命名空間 "admin.\$cmd" 中，另一個位於內部 "TTLMonitor" 任務中。

如果輸出指出許多查詢處於封鎖等待狀態，請參閱[如何尋找並終止長時間執行或封鎖的查詢？](#)

如何判斷一或多個叢集執行個體上 CPU 使用率過高的原因？

下列各節可協助您識別高執行個體 CPU 使用率的原因。結果可能因工作負載而有差異。

- 若要判斷執行個體突然執行緩慢的原因，請參閱[如何確定系統突然運行緩慢的原因？](#)
- 若要識別和終止特定執行個體的長時間執行查詢，請參閱[如何尋找並終止長時間執行或封鎖的查詢？](#)
- 若要了解查詢是否正在進行，請參閱[我如何知道查詢何時正在進行？](#)
- 若要判斷查詢需要長期執行時間的原因，請參閱[如何查看查詢計劃並優化查詢？](#)
- 若要追蹤隨時間演進的長時間執行查詢，請參閱[分析亞馬遜文檔數據庫操作。](#)

根據造成高執行個體 CPU 使用率的原因，執行下列一項或多項動作可能有所幫助。

- 如果主要執行個體顯示高 CPU 使用率，但複本執行個體未顯示同樣情況，則請考慮透過用戶端讀取偏好設定 (例如 `secondaryPreferred`)，將讀取流量分配至所有複本。如需詳細資訊，請參閱 [以複本集形式連接至 Amazon DocumentDB](#)。

使用複本進行讀取時，因為主要執行個體可以處理更多寫入流量，進而有效提升叢集資源的使用效能。來自複本的讀取最終會保持一致。

- 如果寫入工作負載的結果是造成高 CPU 使用率，則將叢集執行個體的大小變更為較大的執行個體類型，將會增加可用於服務工作負載的 CPU 核心數量。如需詳細資訊，請參閱 [執行個體](#) 及 [執行個體類別規格](#)。
- 如果所有叢集執行個體都呈現高 CPU 使用率，且工作負載使用複本進行讀取，則新增更多複本到叢集時，可供讀取流量使用的資源將會增加。如需詳細資訊，請參閱 [將 Amazon DocumentDB 執行個體新增至叢集](#)。

如何確定實例上的打開游標？

連線到 Amazon DocumentDB 執行個體時，您可以使用命令列出該執行個體上開啟的游標。在指定的 Amazon DocumentDB 執行個體上，任何指定時間最多可開啟 4,560 個作用中游標，視執行個體類型而定。我們通常建議關閉不再使用的游標，因為游標會使用執行個體上的資源，且有數目上限的限制。請參閱 [Amazon DocumentDB 配額和限制](#) 以取得特定限制。

```
db.runCommand("listCursors")
```

如何確定當前的 Amazon DocumentDB 引擎版本？

若要判斷您目前的 Amazon DocumentDB 引擎版本，請執行下列命令。

```
db.runCommand({getEngineVersion: 1})
```

此操作的輸出將會如下所示 (JSON 格式)。

```
{ "engineVersion" : "2.x.x", "ok" : 1 }
```

Note

Amazon DocumentDB 3.6 的引擎版本是 1.x.x 和 Amazon DocumentDB 4.0 的引擎版本是 2.x.x。

如何分析索引使用情況並識別未使用的索引？

若要識別指定集合的索引，請執行下列命令：

```
db.collection.getIndexes()
```

若要分析對集合執行作業期間使用的索引數量，可以使用 `collStats` 和 `indexStats` 指令。若要檢視使用索引 (索引掃描) 執行的掃描總數與不使用索引執行的掃描次數 (收集掃描) 的比較，請執行下列命令：

```
db.collection.stats()
```

此命令的輸出包括下列值：

- **idxScans**-使用索引對此集合執行的掃描次數。
- **collScans**-對此集合執行的掃描次數，但不使用索引。這些掃描將涉及一次查看集合中的文檔。
- **lastReset**-這些計數器上次重置的時間 啟動/停止叢集或擴展/縮減執行個體時，會重設此命令提供的統計資料。

在以下命令的輸出中可以找到每個索引使用量的細分。最佳作法是定期識別和移除未使用的索引，以改善效能並降低成本，因為它可以消除用於維護索引的不必要運算、儲存和 I/O。

```
db.collection.aggregate([{$indexStats: {}}]).pretty()
```

此命令的輸出會為集合上建立的每個索引提供下列值：

- **ops**-使用索引的作業數目。如果您的工作負載已執行足夠長的時間，而且您確信您的工作負載處於穩定狀態，ops 的零值表示完全不使用索引。

- **numDocsRead**-使用此索引作業期間讀取的文件數目。
- **since**-自 Amazon DocumentDB 開始收集索引使用狀況統計資料以來的時間，這通常是自上次資料庫重新啟動或維護動作以來的值。
- **size**-此索引的大小 (以位元組為單位)。

下列範例是執行上述命令的範例輸出：

```
{
  "name" : "_id_",
  "key" : {
    "_id" : 1
  },
  "host" : "example-host.com:12345",
  "size" : NumberLong(...),
  "accesses" : {
    "ops" : NumberLong(...),
    "docsRead" : NumberLong(...),
    "since" : ISODate("...")
  },
  "cacheStats" : {
    "blksRead" : NumberLong(...),
    "blksHit" : NumberLong(...),
    "hitRatio" : ...
  }
}
{
  "name" : "x_1",
  "key" : {
    "x" : 1
  },
  "host" : "example-host.com:12345",
  "size" : NumberLong(...),
  "accesses" : {
    "ops" : NumberLong(...),
    "docsRead" : NumberLong(...),
    "since" : ISODate("...")
  },
  "cacheStats" : {
    "blksRead" : NumberLong(...),
    "blksHit" : NumberLong(...),
    "hitRatio" : ...
  }
}
```

```
}
```

若要判斷集合的整體索引大小，請執行下列命令：

```
db.collection.stats()
```

若要刪除未使用的索引，請執行下列命令：

```
db.collection.dropIndex("indexName")
```

如何識別遺失的索引？

您可以使用 [Amazon DocumentDB 效能分析工具](#) 來記錄慢速查詢。在慢速查詢日誌中重複出現查詢，可能表示需要額外的索引來改善該查詢的效能。

您可以找出實用索引的機會，方法是尋找有一或多個階段執行至少一個 COLLSCAN 個階段的長時間執行查詢，這表示它們查詢階段必須讀取集合中的每個文件，以提供對查詢的回應。

下列範例顯示在大型集合上執行的計程車乘車查詢。

```
db.rides.count({"fare.totalAmount":{">10.0}}))
```

為了執行這個範例，查詢必須執行集合掃描 (即讀取集合中的每個文件)，因為該 `fare.totalAmount` 欄位上沒有索引。此查詢的 Amazon DocumentDB 效能分析工具的輸出如下所示：

```
{
  ...
  "cursorExhausted": true,
  "nreturned": 0,
  "responseLength": 0,
  "protocol": "op_query",
  "millis": 300679,
  "planSummary": "COLLSCAN",
  "execStats": {
    "stage": "COLLSCAN",
    "nReturned": "0",
    "executionTimeMillisEstimate": "300678.042"
  },
  "client": "172.31.5.63:53878",
  "appName": "MongoDB Shell",
  "user": "example"
```



```
}
```

為了加快這個例子中的查詢，您可在 `fare.totalAmount` 建立一個索引，如下圖所示。

```
db.rides.createIndex( {"fare.totalAmount": 1}, {background: true} )
```

Note

在前景中建立的索引 (也就是說，如果建立索引時未提供 `{background:true}` 選項) 會採用獨佔寫入鎖定，這樣可以防止應用程式在索引建立完成之前將資料寫入集合。在生產叢集上建立索引時，請注意這項潛在影響。建立索引時，建議設定 `{background:true}`。

一般而言，您會希望在具有高基數 (例如，大量唯一值) 的欄位上建立索引。在低基數的欄位上建立索引可能會導致有大型索引未使用。建立查詢計劃時，Amazon DocumentDB 查詢最佳化工具會考量索引集合的整體大小和選擇性。有時候，您會看到查詢處理器選擇一個 COLLSCAN，即使索引已存在。當查詢處理器預估，相較於掃描整個集合，利用索引並不會產生效能優勢時，就會發生這種情況。如果您想強制查詢處理器利用一個特定索引，您可以使用 `hint()` 運算子，如下圖所示。

```
db.collection.find().hint("indexName")
```

有用的查詢摘要

下列查詢對於在 Amazon DocumentDB 中監控效能和資源使用率非常有用。

- 您可以使用下列命令來檢視特定集合的統計資料，包括作業計數器、快取統計資料、存取統計資料和大小統計資料：

```
db.collection.stats()
```

- 您可以使用下列命令來檢視集合上建立之每個索引的統計資料，包括索引的大小、索引特定快取統計資料，以及索引使用狀況統計資料：

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

- 使用以下查詢來列出所有活動。

```
db.adminCommand({currentOp: 1, $all: 1});
```

- 下面程式碼會列出所有長時間執行或遭封鎖的查詢。

```
db.adminCommand({aggregate: 1,
  pipeline: [{$currentOp: {}},
    {$match: {$or: [{secs_running: {$gt: 10}},
      {WaitState: {$exists: true}}]}]},
  {$project: {_id:0,
    opid: 1,
    secs_running: 1,
    WaitState: 1,
    blockedOn: 1,
    command: 1}}],
  cursor: {}
});
```

- 下列程式碼會終止查詢。

```
db.adminCommand({killOp: 1, op: <opid of running or blocked query>});
```

- 使用下列程式碼來取得系統狀態的彙總檢視。

```
db.adminCommand({aggregate: 1,
  pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
    {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
  "$WaitState"}, count: {$sum: 1}}}],
  cursor: {}
});
```

Amazon DocumentDB 叢集、執行個體和資源管理 API 參考

本節說明可透過 HTTP 存取的叢集、執行個體和資源管理操作 (具有 MongoDB 兼容性) AWS Command Line Interface(AWS CLI) , 或AWS開發套件。您可以使用這些 API 來建立、刪除和修改叢集和執行個體。

Important

這些 API 僅用於管理叢集、執行個體以及相關資源。如需如何連接到執行中資料庫叢集的詳細資訊，請參[開始使用指南](#)。

主題

- [動作](#)
- [資料類型](#)
- [常見錯誤](#)
- [常見參數](#)

動作

支援下列動作 Amazon DocumentDB (with MongoDB compatibility) :

- [AddSourceIdentifierToSubscription](#)
- [AddTagsToResource](#)
- [ApplyPendingMaintenanceAction](#)
- [CopyDBClusterParameterGroup](#)
- [CopyDBClusterSnapshot](#)
- [CreateDBCluster](#)
- [CreateDBClusterParameterGroup](#)
- [CreateDBClusterSnapshot](#)
- [CreateDBInstance](#)
- [CreateDBSubnetGroup](#)
- [CreateEventSubscription](#)

- [CreateGlobalCluster](#)
- [DeleteDBCluster](#)
- [DeleteDBClusterParameterGroup](#)
- [DeleteDBClusterSnapshot](#)
- [DeleteDBInstance](#)
- [DeleteDBSubnetGroup](#)
- [DeleteEventSubscription](#)
- [DeleteGlobalCluster](#)
- [DescribeCertificates](#)
- [DescribeDBClusterParameterGroups](#)
- [DescribeDBClusterParameters](#)
- [DescribeDBClusters](#)
- [DescribeDBClusterSnapshotAttributes](#)
- [DescribeDBClusterSnapshots](#)
- [DescribeDBEngineVersions](#)
- [DescribeDBInstances](#)
- [DescribeDBSubnetGroups](#)
- [DescribeEngineDefaultClusterParameters](#)
- [DescribeEventCategories](#)
- [DescribeEvents](#)
- [DescribeEventSubscriptions](#)
- [DescribeGlobalClusters](#)
- [DescribeOrderableDBInstanceOptions](#)
- [DescribePendingMaintenanceActions](#)
- [FailoverDBCluster](#)
- [ListTagsForResource](#)
- [ModifyDBCluster](#)
- [ModifyDBClusterParameterGroup](#)
- [ModifyDBClusterSnapshotAttribute](#)
- [ModifyDBInstance](#)

- [ModifyDBSubnetGroup](#)
- [ModifyEventSubscription](#)
- [ModifyGlobalCluster](#)
- [RebootDBInstance](#)
- [RemoveFromGlobalCluster](#)
- [RemoveSourceIdentifierFromSubscription](#)
- [RemoveTagsFromResource](#)
- [ResetDBClusterParameterGroup](#)
- [RestoreDBClusterFromSnapshot](#)
- [RestoreDBClusterToPointInTime](#)
- [StartDBCluster](#)
- [StopDBCluster](#)

Amazon DocumentDB 彈性叢集支援下列動作：

- [CopyClusterSnapshot](#)
- [CreateCluster](#)
- [CreateClusterSnapshot](#)
- [DeleteCluster](#)
- [DeleteClusterSnapshot](#)
- [GetCluster](#)
- [GetClusterSnapshot](#)
- [ListClusters](#)
- [ListClusterSnapshots](#)
- [ListTagsForResource](#)
- [RestoreClusterFromSnapshot](#)
- [StartCluster](#)
- [StopCluster](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateCluster](#)

Amazon DocumentDB (with MongoDB compatibility)

支援下列動作Amazon DocumentDB (with MongoDB compatibility) :

- [AddSourceIdentifierToSubscription](#)
- [AddTagsToResource](#)
- [ApplyPendingMaintenanceAction](#)
- [CopyDBClusterParameterGroup](#)
- [CopyDBClusterSnapshot](#)
- [CreateDBCluster](#)
- [CreateDBClusterParameterGroup](#)
- [CreateDBClusterSnapshot](#)
- [CreateDBInstance](#)
- [CreateDBSubnetGroup](#)
- [CreateEventSubscription](#)
- [CreateGlobalCluster](#)
- [DeleteDBCluster](#)
- [DeleteDBClusterParameterGroup](#)
- [DeleteDBClusterSnapshot](#)
- [DeleteDBInstance](#)
- [DeleteDBSubnetGroup](#)
- [DeleteEventSubscription](#)
- [DeleteGlobalCluster](#)
- [DescribeCertificates](#)
- [DescribeDBClusterParameterGroups](#)
- [DescribeDBClusterParameters](#)
- [DescribeDBClusters](#)
- [DescribeDBClusterSnapshotAttributes](#)
- [DescribeDBClusterSnapshots](#)
- [DescribeDBEngineVersions](#)
- [DescribeDBInstances](#)

- [DescribeDBSubnetGroups](#)
- [DescribeEngineDefaultClusterParameters](#)
- [DescribeEventCategories](#)
- [DescribeEvents](#)
- [DescribeEventSubscriptions](#)
- [DescribeGlobalClusters](#)
- [DescribeOrderableDBInstanceOptions](#)
- [DescribePendingMaintenanceActions](#)
- [FailoverDBCluster](#)
- [ListTagsForResource](#)
- [ModifyDBCluster](#)
- [ModifyDBClusterParameterGroup](#)
- [ModifyDBClusterSnapshotAttribute](#)
- [ModifyDBInstance](#)
- [ModifyDBSubnetGroup](#)
- [ModifyEventSubscription](#)
- [ModifyGlobalCluster](#)
- [RebootDBInstance](#)
- [RemoveFromGlobalCluster](#)
- [RemoveSourceIdentifierFromSubscription](#)
- [RemoveTagsFromResource](#)
- [ResetDBClusterParameterGroup](#)
- [RestoreDBClusterFromSnapshot](#)
- [RestoreDBClusterToPointInTime](#)
- [StartDBCluster](#)
- [StopDBCluster](#)

AddSourceIdentifierToSubscription

服務：Amazon DocumentDB (with MongoDB compatibility)

將來源識別符新增至現有的事件通知訂閱。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

SourceIdentifier

要新增之事件來源的識別碼：

- 如果來源類型是執行環境，則DBInstanceIdentifier必須提供。
- 如果來源類型是安全性群組，則DBSecurityGroupName必須提供。
- 如果來源類型是參數群組，則DBParameterGroupName必須提供。
- 如果來源類型是快照，則DBSnapshotIdentifier必須提供。

類型：字串

必要：是

SubscriptionName

您要新增來源識別碼的 Amazon DocumentDB 事件通知訂閱的名稱。

類型：字串

必要：是

回應元素

服務會傳回下列元素。

EventSubscription

有關您已訂閱之事件的詳細資訊。

類型：[EventSubscription](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

SourceNotFound

找不到要求的來源。

HTTP 狀態碼：404

SubscriptionNotFound

訂閱名稱不存在。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

AddTagsToResource

服務：Amazon DocumentDB (with MongoDB compatibility)

將中繼資料標籤新增至 Amazon DocumentDB 源。您可以將這些標籤與成本分配報告搭配使用，以追蹤與 Amazon DocumentDB 資源相關聯的成本，或在 Amazon DocumentDB AWS Identity and Access Management (IAM) 政策中的Condition陳述式中追蹤成本。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

ResourceName

該標籤被添加到 Amazon DocumentDB 資源。這個值是一個 Amazon 資源名稱。

類型：字串

必要：是

標籤. N

要分配給 Amazon DocumentDB 資源的標籤。

類型：[Tag](#) 物件陣列

必要：是

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

DBClusterIdentifier不參考現有叢集。

HTTP 狀態碼：404

DBInstanceNotFound

DBInstanceIdentifier不會參考現有的執行個體。

HTTP 狀態碼：404

DBSnapshotNotFound

DBSnapshotIdentifier 不參考現有的快照。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ApplyPendingMaintenanceAction

服務：Amazon DocumentDB (with MongoDB compatibility)

將擱置中的維護動作套用至資源 (例如，套用至 Amazon DocumentDB 執行個體)。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

ApplyAction

要套用至此資源的待處理維護動作。

有效值：system-update、db-upgrade

類型：字串

必要：是

OptInType

指定選擇加入要求類型或復原選擇加入要求的值。無法復原 immediate 類型的選擇使用請求。

有效值：

- immediate - 立即套用維護動作。
- next-maintenance - 在下次資源維護時段期間套用維護動作。
- undo-opt-in - 取消任何現有的 next-maintenance 選擇使用請求。

類型：字串

必要：是

ResourceIdentifier

要套用待處理維護動作之資源的 Amazon Resource Name (ARN)。

類型：字串

必要：是

回應元素

服務會傳回下列元素。

ResourcePendingMaintenanceActions

代表的輸出[ApplyPendingMaintenanceAction](#)。

類型：[ResourcePendingMaintenanceActions](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

ResourceNotFoundFault

找不到指定的資源 ID。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)

- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CopyDBClusterParameterGroup

服務：Amazon DocumentDB (with MongoDB compatibility)

複製指定的叢集參數群組。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

SourceDBClusterParameterGroupIdentifier

來源叢集參數群組的識別碼或 Amazon 資源名稱 (ARN)。

約束：

- 必須指定有效的叢集參數群組。
- 如果來源叢集參數群組與副本AWS 區域相同，請指定有效的參數群組識別碼；例如my-db-cluster-param-group，或有效的 ARN。
- 如果來源參數群組與副本AWS 區域不同，請指定有效的叢集參數群組 ARN，arn:aws:rds:us-east-1:123456789012:sample-cluster:sample-parameter-group例如。

類型：字串

必要：是

TargetDBClusterParameterGroupDescription

複製之叢集參數群組的說明。

類型：字串

必要：是

TargetDBClusterParameterGroupIdentifier

複製之叢集參數群組的識別碼。

約束：

- 不得為無效、空白或留空。
- 必須包含 1 到 255 個字母、數字或連字號。
- 第一個字元必須是字母。

- 不能以連字號結尾或連續包含兩個連字號。

範例：my-cluster-param-group1

類型：字串

必要：是

標籤. N

要指定給參數群組的標籤。

類型：[Tag](#) 物件陣列

必要：否

回應元素

服務會傳回下列元素。

DBClusterParameterGroup

叢集參數群組的詳細資訊。

類型：[DBClusterParameterGroup](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBParameterGroupAlreadyExists

具有相同名稱的參數群組已存在。

HTTP 狀態碼：400

DBParameterGroupNotFound

DBParameterGroupName不參考現有的參數群組。

HTTP 狀態碼：404

DBParameterGroupQuotaExceeded

此要求會導致您超過允許的參數群組數目。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CopyDBClusterSnapshot

服務：Amazon DocumentDB (with MongoDB compatibility)

複製叢集的快照。

若要從共用手動叢集快照複製叢集快照，SourceDBClusterSnapshotIdentifier必須是共用叢集快照的 Amazon 資源名稱 (ARN)。您只能在相同的 AWS 區域中複製共用資料庫叢集快照，無論是否加密。

若要在進行複製作業之後取消複製作業，請刪除叢集快照處於複製狀態TargetDBClusterSnapshotIdentifier時所識別的目標叢集快照。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

SourceDBClusterSnapshotIdentifier

要複製之叢集快照的識別碼。此參數不區分大小寫。

約束：

- 必須指定可用狀態下的有效系統快照。
- 如果來源快照與副本相AWS區域同，請指定有效的快照識別碼。
- 如果來源快照與副本AWS區域不同，請指定有效的叢集快照 ARN。

範例：my-cluster-snapshot1

類型：字串

必要：是

TargetDBClusterSnapshotIdentifier

要從來源叢集快照建立之新叢集快照的識別碼。此參數不區分大小寫。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：my-cluster-snapshot2

類型：字串

必要：是

CopyTags

設定為true可將來源叢集快照中的所有標籤複製到目標叢集快照，否則false。預設值為false。

類型：布林值

必要：否

KmsKeyId

加密叢集快照的AWS KMS金鑰識別碼。AWS KMS 金鑰 ID 是 AWS KMS 加密金鑰的 Amazon Resource Name (ARN)、AWS KMS 金鑰識別符或 AWS KMS 金鑰別名。

如果您從中複製加密的叢集快照AWS 帳戶，您可以指定一個值，以KmsKeyId便使用新的AWS KMS加密金鑰加密副本。如果未指定的值KmsKeyId，則會使用與來源叢集快照相同的AWS KMS金鑰來加密叢集快照的副本。

如果您複製從另一個叢集共用的加密叢集快照AWS 帳戶，則必須指定的值KmsKeyId。

若要將加密的叢集快照複製KmsKeyId到另一個AWS 區域，請設定為要用來加密目的地區域中叢集快照副本的AWS KMS金鑰 ID。AWS KMS加密金鑰專屬於它AWS 區域們所建立的金鑰，您無法使用另一個AWS 區域金鑰的加密金鑰AWS 區域。

如果您複製未加密的叢集快照並指定KmsKeyId參數的值，則會傳回錯誤。

類型：字串

必要：否

PreSignedUrl

包含針對 CopyDBClusterSnapshot API 動作的簽章第 4 版已簽署要求的 URL，其中包AWS 區域含要複製的來源叢集快照集。從另一個叢集快照複製叢集快照時，您必須使用PreSignedUrl參數AWS 區域。

如果您使用的是 AWS SDK 工具或AWS CLI，您可以指定 SourceRegion (或 --source-regionAWS CLI)，而不是PreSignedUrl手動指定。指定SourceRegion自動產生預先簽署的 URL，此 URL 是可在來源中執行之作業的有效要求。AWS 區域

預先簽署的 URL 必須是 CopyDBClusterSnapshot API 動作的有效要求，該動作可在包含要複製的叢集快照的來源AWS 區域中執行。預先簽署的 URL 請求必須包含下列參數值：

- SourceRegion-包含要複製之快照的區域 ID。
- SourceDBClusterSnapshotIdentifier-要複製之加密叢集快照的識別碼。此識別符必須是來源 AWS 區域的 Amazon Resource Name (ARN) 格式。例如，如果您要從 us-east-1 複製加密的叢集快照AWS 區域，則您的SourceDBClusterSnapshotIdentifier外觀如下所示：。arn:aws:rds:us-east-1:12345678012:sample-cluster:sample-cluster-snapshot
- TargetDBClusterSnapshotIdentifier-要建立之新叢集快照的識別碼。此參數不區分大小寫。

類型：字串

必要：否

標籤. N

要指派給叢集快照的標籤。

類型：[Tag](#) 物件陣列

必要：否

回應元素

服務會傳回下列元素。

DBClusterSnapshot

叢集快照的詳細資訊。

類型：[DBClusterSnapshot](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterSnapshotAlreadyExistsFault

您已經擁有具有指定識別碼的叢集快照。

HTTP 狀態碼：400

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier不會參考現有的叢集快照。

HTTP 狀態碼：404

InvalidDBClusterSnapshotStateFault

提供的值不是有效的叢集快照狀態。

HTTP 狀態碼：400

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

KMSKeyNotAccessibleFault

存取AWS KMS金鑰時發生錯誤。

HTTP 狀態碼：400

SnapshotQuotaExceeded

該請求會導致您超過允許的快照數量。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)

- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CreateDBCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

建立新的 Amazon DocumentDB 叢集。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

叢集的識別碼，此參數是以小寫字母字串的形式儲存。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：my-cluster

類型：字串

必要：是

Engine

要用於此叢集的資料庫引擎名稱。

有效值：docdb

類型：字串

必要：是

AvailabilityZones。 AvailabilityZone.N

Amazon EC2 可用區域的清單，在這些可用區域中可以建立叢集中的執行個體。

類型：字串陣列

必要：否

BackupRetentionPeriod

自動備份保留的天數。您必須指定一個值 (最小值為 1)。

預設：1

約束：

- 該值必須介於 1 到 35 之間。

類型：整數

必要：否

DBClusterParameterGroupName

要與此叢集建立關聯的叢集參數群組名稱。

類型：字串

必要：否

DBSubnetGroupName

要與此叢集建立關聯的子網路群組。

限制條件：必須符合現有 DBSubnetGroup 的名稱。絕不能為預設值。

範例：mySubnetgroup

類型：字串

必要：否

DeletionProtection

指定是否可以刪除此叢集。如果啟 DeletionProtection 用，除非已修改並 DeletionProtection 停用叢集，否則無法刪除叢集。DeletionProtection 防止叢集遭到意外刪除。

類型：布林值

必要：否

EnableCloudwatchLogsExports. 成員.

必須啟用匯出至 Amazon CloudWatch 日誌的記錄類型清單。您可以啟用稽核日誌或分析工具日誌。如需詳細資訊，請參閱[稽核 Amazon DocumentDB 事件](#)和[分析 Amazon DocumentDB 操作](#)。

類型：字串陣列

必要：否

EngineVersion

要使用的資料庫引擎版本號碼。--engine-version 預設為最新的主要引擎版本。對於生產工作負載，我們建議使用預定的主要引擎版本明確宣告此參數。

類型：字串

必要：否

GlobalClusterIdentifier

新全域叢集的叢集識別碼。

類型：字串

長度限制：長度下限為 1。長度上限為 255。

模式：[A-Za-z][0-9A-Za-z-:._]*

必要：否

KmsKeyId

加密叢集的 AWS KMS 金鑰識別碼。

AWS KMS 金鑰識別碼是 AWS KMS 加密金鑰的 Amazon Resource Name (ARN)。若您使用擁有用來加密新叢集 AWS 帳戶 加密金鑰的相同 AWS KMS 建立叢集，您可以改為使用 AWS KMS 金鑰別名，而非 AWS KMS 加密金鑰的 ARN。

若加密金鑰並未在 KmsKeyId 中指定：

- 如果 StorageEncrypted 參數是 true，則 Amazon DocumentDB 會使用您的預設加密金鑰。

AWS KMS 會為 AWS 帳戶 建立預設加密金鑰。您的 AWS 帳戶 在每個 AWS 區域 各有不同的預設加密金鑰。

類型：字串

必要：否

MasterUsername

叢集主要使用者的名稱。

約束:

- 必須為 1 到 63 個字母或數字。
- 第一個字元必須是字母。
- 不能是所選擇資料庫引擎的保留字。

類型：字串

必要：否

MasterUserPassword

主要資料庫使用者的密碼。此密碼可包含任何可印出的 ASCII 字元，但斜線 (/)、雙引號 (") 或 "at" 符號 (@) 則除外。

限制條件：必須包含 8 到 100 個字元。

類型：字串

必要：否

Port

叢集中執行處理接受連線的連接埠號碼。

類型：整數

必要：否

PreferredBackupWindow

每日的時間範圍，若使用 BackupRetentionPeriod 參數啟用自動化備份，則會在此期間建立自動化備份。

預設是從各 AWS 區域 8 小時時段內隨機選取的 30 分鐘時段。

約束:

- 格式必須為 hh24:mi-hh24:mi。
- 必須以國際標準時間 (UTC) 表示。
- 不得和慣用的維護時段衝突。
- 必須至少 30 分鐘。

類型：字串

必要：否

PreferredMaintenanceWindow

每週可能進行系統維護的時段，以國際標準時間 (UTC) 表示。

格式：ddd:hh24:mi-ddd:hh24:mi

預設是從各 AWS 區域 8 小時時段內隨機選取的 30 分鐘時段，並隨機發生在一週內的某一天。

有效日：星期一、星期二、星期三、星期四、星期五、星期六、星期日

限制條件：必須至少是 30 分鐘的時段。

類型：字串

必要：否

PreSignedUrl

目前不支援。

類型：字串

必要：否

StorageEncrypted

指定是否要加密叢集。

類型：布林值

必要：否

StorageType

和資料庫叢集相關聯的儲存類型。

如需 Amazon DocumentDB 叢集的儲存類型的相關資訊，請參閱 Amazon DocumentDB 開發人員指南中的叢集儲存組態。

儲存類型的有效值-standard | iopt1

預設值為 standard

Note

當您創建一個 DocumentDB 數據庫集群與存儲類型設置為 `iopt1`，存儲類型在響應返回。當您將其設定為時，不會傳回儲存區類型 `standard`。

類型：字串

必要：否

標籤. N

要指派給叢集的標籤。

類型：[Tag](#) 物件陣列

必要：否

VpcSecurityGroupIds。 VpcSecurityGroupId.N

與此叢集關聯的 EC2 VPC 安全群組清單。

類型：字串陣列

必要：否

回應元素

服務會傳回下列元素。

DBCluster

叢集的詳細資訊。

類型：[DBCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterAlreadyExistsFault

您已經擁有具有指定識別碼的叢集。

HTTP 狀態碼：400

DBClusterNotFoundFault

DBClusterIdentifier不參考現有叢集。

HTTP 狀態碼：404

DBClusterParameterGroupNotFound

DBClusterParameterGroupName不參考現有的叢集參數群組。

HTTP 狀態碼：404

DBClusterQuotaExceededFault

無法建立叢集，因為您已達到叢集允許的最大配額。

HTTP 狀態碼：403

DBInstanceNotFound

DBInstanceIdentifier不會參考現有的執行個體。

HTTP 狀態碼：404

DBSubnetGroupDoesNotCoverEnoughAZs

除非只有一個可用區域，否則子網路群組中的子網路應至少涵蓋兩個可用區域。

HTTP 狀態碼：400

DBSubnetGroupNotFoundFault

DBSubnetGroupName不參照現有的子網路群組。

HTTP 狀態碼：404

GlobalClusterNotFoundFault

GlobalClusterIdentifier不會參照現有的全域叢集。

HTTP 狀態碼：404

InsufficientStorageClusterCapacity

沒有足夠的儲存空間可用於目前動作。您可以透過將子網路群組更新為使用具有更多儲存空間的不同可用區域來解決此錯誤。

HTTP 狀態碼：400

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

InvalidDBSubnetGroupStateFault

子網路群組無法刪除，因為它正在使用中。

HTTP 狀態碼：400

InvalidGlobalClusterStateFault

當群集處於此狀態時，請求的操作無法執行。

HTTP 狀態碼：400

InvalidSubnet

要求的子網路無效，或要求多個子網路並非全部位於一般虛擬私人雲端 (VPC) 中。

HTTP 狀態碼：400

InvalidVPCNetworkStateFault

子網路群組在建立之後，並不會涵蓋所有可用區域，因為已進行變更。

HTTP 狀態碼：400

KMSKeyNotAccessibleFault

存取AWS KMS金鑰時發生錯誤。

HTTP 狀態碼：400

StorageQuotaExceeded

該請求會導致您超過所有執行個體允許的可用儲存空間量。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CreateDBClusterParameterGroup

服務：Amazon DocumentDB (with MongoDB compatibility)

建立新的叢集參數群組。

叢集參數群組中的參數會套用到叢集中的所有執行個體。

最初會為叢集中執行個體所使用的資料庫引擎，使用預設參數建立叢集參數群組。在 Amazon DocumentDB 中，您無法直接對 default.docdb3.6 叢集參數群組進行修改。如果 Amazon DocumentDB 叢集使用預設叢集參數群組，而您想要修改其中的值，則必須先 [建立新參數群組或複製現有參數群組](#)、修改它，然後將修改後的參數群組套用至叢集。若要讓新叢集參數群組和相關設定生效，您必須重新啟動叢集中的執行個體而不進行容錯移轉。如需詳細資訊，請參閱 [修改 Amazon DocumentDB 叢集參數群組](#)。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterParameterGroupName

叢集參數群組的名稱。

約束：

- 不得與現有 DBClusterParameterGroup 的名稱相符。

Note

此值會以小寫字母字串的形式儲存。

類型：字串

必要：是

DBParameterGroupFamily

叢集參數群組系列的名稱。

類型：字串

必要：是

Description

叢集參數群組的描述。

類型：字串

必要：是

標籤. N

要指派給叢集參數群組的標籤。

類型：[Tag](#) 物件陣列

必要：否

回應元素

服務會傳回下列元素。

DBClusterParameterGroup

叢集參數群組的詳細資訊。

類型：[DBClusterParameterGroup](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBParameterGroupAlreadyExists

具有相同名稱的參數群組已存在。

HTTP 狀態碼：400

DBParameterGroupQuotaExceeded

此要求會導致您超過允許的參數群組數目。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CreateDBClusterSnapshot

服務：Amazon DocumentDB (with MongoDB compatibility)

建立叢集的快照。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBClusterIdentifier

要為其建立快照的叢集識別碼。此參數不區分大小寫。

約束：

- 必須符合現有 `DBCluster` 的識別碼。

範例：my-cluster

類型：字串

必要：是

DBClusterSnapshotIdentifier

叢集快照的識別碼。此參數是以小寫字母字串的形式儲存。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：my-cluster-snapshot1

類型：字串

必要：是

標籤. N

要指派給叢集快照的標籤。

類型：[Tag](#) 物件陣列

必要：否

回應元素

服務會傳回下列元素。

DBClusterSnapshot

叢集快照的詳細資訊。

類型：[DBClusterSnapshot](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

DBClusterIdentifier不參考現有叢集。

HTTP 狀態碼：404

DBClusterSnapshotAlreadyExistsFault

您已經擁有具有指定識別碼的叢集快照。

HTTP 狀態碼：400

InvalidDBClusterSnapshotStateFault

提供的值不是有效的叢集快照狀態。

HTTP 狀態碼：400

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

SnapshotQuotaExceeded

該請求會導致您超過允許的快照數量。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CreateDBInstance

服務：Amazon DocumentDB (with MongoDB compatibility)

建立新執行個體。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

執行個體所屬叢集的識別碼。

類型：字串

必要：是

DBInstanceClass

執行個體的運算與記憶體容量，例如 `db.r5.large`。

類型：字串

必要：是

DBInstanceIdentifier

執行個體識別碼。此參數是以小寫字母字串的形式儲存。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：`mydbinstance`

類型：字串

必要：是

Engine

要為此執行個體使用的資料庫引擎名稱。

有效值：docdb

類型：字串

必要：是

AutoMinorVersionUpgrade

此參數不適用於 Amazon DocumentDB。無論設定的值為何，Amazon DocumentDB 都不會執行次要版本升級。

預設：false

類型：布林值

必要：否

AvailabilityZone

在其中建立執行個體的 Amazon EC2 可用區域。

預設：端點 AWS 區域 中系統隨機選擇的可用區域。

範例：us-east-1d

類型：字串

必要：否

CACertificateIdentifier

用於資料庫執行個體伺服器憑證的 CA 憑證識別碼。

如需詳細資訊，請參閱 [Amazon DocumentDB 開發人員指南中的更新您的 Amazon 文件資料庫 TLS 憑證和加密傳輸中的資料](#)。

類型：字串

必要：否

CopyTagsToSnapshot

值，指示是否將標籤從資料庫執行個體複製到資料庫執行個體快照。根據預設，不會複製標籤。

類型：布林值

必要：否

EnablePerformanceInsights

指示是否啟用資料庫執行個體績效詳情的值。如需詳細資訊，請參閱[使用 Amazon 績效詳情](#)。

類型：布林值

必要：否

PerformanceInsightsKMSKeyId

用於 Performance Insights 資料加密的 AWS KMS 金鑰識別符。

金AWS KMS鑰識別碼是 KMS 金鑰的金鑰 ARN、金鑰識別碼、別名 ARN 或別名。

如果您沒有為 PerformanceInsights KMS 指定值KeyId，則 Amazon DocumentDB 會使用您的預設 KMS 金鑰。您的 Amazon Web Services 帳戶有一個默認的 KMS 密鑰。您的 Amazon Web Services 帳戶對每個亞 Amazon Web Services 區域都有不同的預設 KMS 金鑰。

類型：字串

必要：否

PreferredMaintenanceWindow

每週可能進行系統維護的時間範圍，以國際標準時間 (UTC) 表示。

格式：ddd:hh24:mi-ddd:hh24:mi

預設是從各 AWS 區域 8 小時時段內隨機選取的 30 分鐘時段，並隨機發生在一週內的某一天。

有效日：星期一、星期二、星期三、星期四、星期五、星期六、星期日

限制條件：必須至少是 30 分鐘的時段。

類型：字串

必要：否

PromotionTier

一個值，指定 Amazon DocumentDB 複本在現有主執行個體發生故障後提升為主執行個體的順序。

預設：1

有效值：0 至 15

類型：整數

必要：否

標籤. N

要指派給執行個體的標籤。一個執行個體最多可以指派 10 個標籤。

類型：[Tag](#) 物件陣列

必要：否

回應元素

服務會傳回下列元素。

DBInstance

執行個體的詳細資訊。

類型：[DBInstance](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AuthorizationNotFound

指定的 CIDR IP 或 Amazon EC2 安全群組未獲得指定安全群組的授權。

Amazon DocumentDB 也可能無法獲得使用 IAM 代表您執行必要動作的授權。

HTTP 狀態碼：404

DBClusterNotFoundFault

`DBClusterIdentifier` 不參照現有的叢集。

HTTP 狀態碼：404

DBInstanceAlreadyExists

您已經擁有具有指定識別碼的執行個體。

HTTP 狀態碼：400

DBParameterGroupNotFound

DBParameterGroupName不參考現有的參數群組。

HTTP 狀態碼：404

DBSecurityGroupNotFound

DBSecurityGroupName不參考現有的安全性群組。

HTTP 狀態碼：404

DBSubnetGroupDoesNotCoverEnoughAZs

除非只有一個可用區域，否則子網路群組中的子網路應至少涵蓋兩個可用區域。

HTTP 狀態碼：400

DBSubnetGroupNotFoundFault

DBSubnetGroupName不參照現有的子網路群組。

HTTP 狀態碼：404

InstanceQuotaExceeded

該請求會導致您超過允許的實例數量。

HTTP 狀態碼：400

InsufficientDBInstanceCapacity

指定的執行個體類別在指定的可用區域中無法使用。

HTTP 狀態碼：400

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidSubnet

要求的子網路無效，或要求多個子網路並非全部位於一般虛擬私人雲端 (VPC) 中。

HTTP 狀態碼：400

InvalidVPCNetworkStateFault

子網路群組在建立之後，並不會涵蓋所有可用區域，因為已進行變更。

HTTP 狀態碼：400

KMSKeyNotAccessibleFault

存取AWS KMS金鑰時發生錯誤。

HTTP 狀態碼：400

StorageQuotaExceeded

該請求會導致您超過所有執行個體允許的可用儲存空間量。

HTTP 狀態碼：400

StorageTypeNotSupported

指定的儲存體StorageType無法與資料庫執行個體相關聯。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CreateDBSubnetGroup

服務：Amazon DocumentDB (with MongoDB compatibility)

建立新的子網路群組。子網路群組必須至少包含一個子網路，位於中的兩個可用區域。AWS 區域

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBSubnetGroupDescription

子網路群組的描述。

類型：字串

必要：是

DBSubnetGroupName

子網路群組的名稱。此值會以小寫字母字串的形式儲存。

限制條件：包含的內容絕不能超過 255 個字母、數字、句號、底線、空格或連字號。絕不能為預設值。

範例：mySubnetgroup

類型：字串

必要：是

SubnetIds。 SubnetIdentifier.N

子網路群組的 Amazon EC2 子網路 ID。

類型：字串陣列

必要：是

標籤. N

要指派給子網路群組的標籤。

類型：[Tag](#) 物件陣列

必要：否

回應元素

服務會傳回下列元素。

DBSubnetGroup

子網路群組的詳細資訊。

類型：[DBSubnetGroup](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBSubnetGroupAlreadyExists

DBSubnetGroupName現有子網路群組已在使用中。

HTTP 狀態碼：400

DBSubnetGroupDoesNotCoverEnoughAZs

除非只有一個可用區域，否則子網路群組中的子網路應至少涵蓋兩個可用區域。

HTTP 狀態碼：400

DBSubnetGroupQuotaExceeded

此要求會造成您超過允許的子網路群組數目。

HTTP 狀態碼：400

DBSubnetQuotaExceededFault

此要求會導致您超過子網路群組中允許的子網路數目。

HTTP 狀態碼：400

InvalidSubnet

要求的子網路無效，或要求多個子網路並非全部位於一般虛擬私人雲端 (VPC) 中。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CreateEventSubscription

服務：Amazon DocumentDB (with MongoDB compatibility)

創建一個 Amazon DocumentDB 事件通知訂閱。此動作需要使用 Amazon DocumentDB 主控台、Amazon SNS 主控台或 Amazon SNS API 建立的主題亞馬遜資源名稱 (ARN)。若要透過 Amazon SNS 取得 ARN，您必須在 Amazon SNS 中建立主題並訂閱該主題。ARN 會顯示在 Amazon SNS 主控台中。

您可以指定要通知的 source (SourceType) 類型。您也可以提供可觸發事件的 Amazon DocumentDB 來源清單 (SourceIds)，並且可以針對您要收到通知的事件提供事件類別清單 (EventCategories)。例如，您可以指定 SourceType = db-instance、SourceIds = mydbinstance1, mydbinstance2 和 EventCategories = Availability, Backup。

如果您同時指定 SourceType 和 SourceIds (例如 SourceType = db-instance 和 SourceIdentifier = myDBInstance1)，系統會通知您指定來源的所有 db-instance 事件。如果您指定 SourceType 但未指定 SourceIdentifier，您會收到有關所有 Amazon DocumentDB 來源之該來源類型事件的通知。如果您未指定 SourceType 或 SourceIdentifier，系統會通知您從屬於您客戶帳戶的所有 Amazon DocumentDB 來源產生的事件。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

SnsTopicArn

針對事件通知所建立 SNS 主題的 Amazon Resource Name (ARN)。當您建立主題並訂閱主題時，Amazon SNS 會建立 ARN。

類型：字串

必要：是

SubscriptionName

訂閱的名稱。

限制條件：名稱必須少於 255 個字元。

類型：字串

必要：是

Enabled

Boolean 值；設定true為啟動訂閱，設定false為建立訂閱但未啟用訂閱。

類型：布林值

必要：否

EventCategories。 EventCategory.N

您要訂閱的事件類別清單。 SourceType

類型：字串陣列

必要：否

SourceIds。 SourceId.N

傳回事件的事件來源識別碼清單。如未指定，回應中會包含所有來源。識別碼必須以字母開頭，只能包含 ASCII 字母、數字和連字號，而且不得以連字號結尾，或連續包含兩個連字號。

約束:

- 如果SourceIds有提供，也SourceType必須提供。
- 如果來源類型是執行環境，則DBInstanceIdentifier必須提供。
- 如果來源類型是安全性群組，則DBSecurityGroupName必須提供。
- 如果來源類型是參數群組，則DBParameterGroupName必須提供。
- 如果來源類型是快照，則DBSnapshotIdentifier必須提供。

類型：字串陣列

必要：否

SourceType

產生事件的來源類型。例如，如果您想要收到執行處理所產生之事件的通知，您可以將此參數設定為db-instance。若沒有指定此值，則會傳回所有事件。

有效值：db-instance、db-cluster、db-parameter-group、db-security-group、db-cluster-snapshot

類型：字串

必要：否

標籤. N

要指派給活動訂閱的標籤。

類型：[Tag](#) 物件陣列

必要：否

回應元素

服務會傳回下列元素。

EventSubscription

有關您已訂閱之事件的詳細資訊。

類型：[EventSubscription](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

EventSubscriptionQuotaExceeded

你已達到活動訂閱數目上限。

HTTP 狀態碼：400

SNSInvalidTopic

Amazon SNS 已回應指定主題存在問題。

HTTP 狀態碼：400

SNSNoAuthorization

您沒有發佈至 SNS 主題 Amazon 資源名稱 (ARN) 的權限。

HTTP 狀態碼：400

SNSTopicArnNotFound

SNS 主題 Amazon 資源名稱 (ARN) 不存在。

HTTP 狀態碼：404

SourceNotFound

找不到要求的來源。

HTTP 狀態碼：404

SubscriptionAlreadyExist

提供的訂閱名稱已存在。

HTTP 狀態碼：400

SubscriptionCategoryNotFound

提供的類別不存在。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

CreateGlobalCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

建立可跨越多個多個的 Amazon DocumentDB 全域叢集。AWS 區域全域叢集包含一個具有讀寫功能的主要叢集，並提供唯讀次要叢集。全域叢集使用以儲存為基礎的快速複寫，跨延遲時間少於一秒的區域，使用專用基礎架構，而不會影響工作負載的效能。

您可以建立一個最初為空的全域叢集，然後在其中新增主叢集和次要叢集。或者，您可以在建立作業期間指定現有叢集，此叢集就會成為全域叢集的主要叢集。

Note

此動作僅適用於 Amazon DocumentDB 叢集。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

GlobalClusterIdentifier

新全域叢集的叢集識別碼。

類型：字串

長度限制：長度下限為 1。長度上限為 255。

模式：[A-Za-z][0-9A-Za-z-:._]*

必要：是

DatabaseName

您資料庫的名稱最長可達 64 個英數字元。如果您未提供名稱，Amazon DocumentDB 將不會在您建立的全域叢集中建立資料庫。

類型：字串

必要：否

DeletionProtection

新全域叢集的刪除保護設定。啟用刪除保護時，無法刪除全域叢集。

類型：布林值

必要：否

Engine

要用於此叢集的資料庫引擎名稱。

類型：字串

必要：否

EngineVersion

全域叢集的引擎版本。

類型：字串

必要：否

SourceDBClusterIdentifier

Amazon 資源名稱 (ARN) 用作全域叢集的主要叢集。此為選用參數。

類型：字串

必要：否

StorageEncrypted

新全域叢集的儲存區加密設定。

類型：布林值

必要：否

回應元素

服務會傳回下列元素。

GlobalCluster

代表 Amazon 文件資料庫全域叢集的資料類型。

類型：[GlobalCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

`DBClusterIdentifier`不參考現有叢集。

HTTP 狀態碼：404

GlobalClusterAlreadyExistsFault

`GlobalClusterIdentifier` 已存在。選擇新的全域叢集識別碼 (唯一名稱) 以建立新的全域叢集。

HTTP 狀態碼：400

GlobalClusterQuotaExceededFault

此帳戶的全域叢集數目已達到允許的最大值。

HTTP 狀態碼：400

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)

- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DeleteDBCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

刪除先前佈建的叢集。刪除叢集時，該叢集的所有自動備份都會遭到刪除，且無法復原。不會刪除指定叢集的手動資料庫叢集快照。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

要刪除之叢集的叢集識別碼。此參數不區分大小寫。

約束：

- 必須符合現有的DBClusterIdentifier。

類型：字串

必要：是

FinalDBSnapshotIdentifier

設為時SkipFinalSnapshot所建立之新叢集快照的叢集快照識別碼false。

Note

指定此參數並將SkipFinalShapshot參數設定為會true導致錯誤。

約束：

- 必須介於 1 至 255 個字母、數字或連字號之間。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

類型：字串

必要：否

SkipFinalSnapshot

決定是否在刪除叢集之前建立最終叢集快照。如果true指定，則不會建立叢集快照。如果false已指定，則會在刪除資料庫叢集之前建立叢集快照。

Note

如果SkipFinalSnapshot是false，則必須指定FinalDBSnapshotIdentifier參數。

預設：false

類型：布林值

必要：否

回應元素

服務會傳回下列元素。

DBCluster

叢集的詳細資訊。

類型：[DBCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

DBClusterIdentifier不參照現有的叢集。

HTTP 狀態碼：404

DBClusterSnapshotAlreadyExistsFault

您已經擁有具有指定識別碼的叢集快照。

HTTP 狀態碼：400

InvalidDBClusterSnapshotStateFault

提供的值不是有效的叢集快照狀態。

HTTP 狀態碼：400

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

SnapshotQuotaExceeded

該請求會導致您超過允許的快照數量。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DeleteDBClusterParameterGroup

服務：Amazon DocumentDB (with MongoDB compatibility)

刪除指定的叢集參數群組。要刪除的叢集參數群組不能與任何叢集相關聯。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterParameterGroupName

叢集參數群組的名稱。

約束：

- 必須是現有叢集參數群組的名稱。
- 您無法刪除預設叢集參數群組。
- 無法與任何叢集關聯。

類型：字串

必要：是

錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

DBParameterGroupNotFound

DBParameterGroupName不參考現有的參數群組。

HTTP 狀態碼：404

InvalidDBParameterGroupState

參數群組正在使用中，或處於無效狀態。如果您嘗試刪除參數群組，則無法在參數群組處於此狀態時將其刪除。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DeleteDBClusterSnapshot

服務：Amazon DocumentDB (with MongoDB compatibility)

刪除叢集快照。如果正在複製快照，即會終止複製操作。

Note

叢集快照必須處於要刪除的available狀態。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBClusterSnapshotIdentifier

要刪除之叢集快照的識別碼。

條件約束：必須是available狀態中現有叢集快照的名稱。

類型：字串

必要：是

回應元素

服務會傳回下列元素。

DBClusterSnapshot

叢集快照的詳細資訊。

類型：[DBClusterSnapshot](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier不會參考現有的叢集快照。

HTTP 狀態碼：404

InvalidDBClusterSnapshotStateFault

提供的值不是有效的叢集快照狀態。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DeleteDBInstance

服務：Amazon DocumentDB (with MongoDB compatibility)

刪除先前佈建的執行個體。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBInstanceIdentifier

要刪除之執行個體的執行個體識別碼。此參數不區分大小寫。

約束：

- 必須與現有執行個體的名稱相符。

類型：字串

必要：是

回應元素

服務會傳回下列元素。

DBInstance

執行個體的詳細資訊。

類型：[DBInstance](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

DBInstanceNotFound

DBInstanceIdentifier 不會參考現有的執行個體。

HTTP 狀態碼：404

DBSnapshotAlreadyExists

DBSnapshotIdentifier 現有快照已在使用中。

HTTP 狀態碼：400

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

SnapshotQuotaExceeded

該請求會導致您超過允許的快照數量。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DeleteDBSubnetGroup

服務：Amazon DocumentDB (with MongoDB compatibility)

刪除子網路群組。

Note

指定的資料庫子網路群組不能和任何資料庫執行個體建立關聯。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBSubnetGroupName

要刪除的資料庫子網路群組名稱。

Note

您無法刪除預設子網路群組。

約束：

必須符合現有 DBSubnetGroup 的名稱。絕不能為預設值。

範例：mySubnetgroup

類型：字串

必要：是

錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

DBSubnetGroupNotFoundFault

DBSubnetGroupName 不參照現有的子網路群組。

HTTP 狀態碼：404

InvalidDBSubnetGroupStateFault

子網路群組無法刪除，因為它正在使用中。

HTTP 狀態碼：400

InvalidDBSubnetStateFault

子網路未處於可用狀態。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DeleteEventSubscription

服務：Amazon DocumentDB (with MongoDB compatibility)

刪除 Amazon DocumentDB 事件通知訂閱。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

SubscriptionName

您要刪除的 Amazon DocumentDB 事件通知訂閱的名稱。

類型：字串

必要：是

回應元素

服務會傳回下列元素。

EventSubscription

有關您已訂閱之事件的詳細資訊。

類型：[EventSubscription](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

InvalidEventSubscriptionState

其他人可能正在修改訂閱。等待幾秒鐘，然後再試一次。

HTTP 狀態碼：400

SubscriptionNotFound

訂閱名稱不存在。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DeleteGlobalCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

刪除全域叢集。在嘗試刪除全域叢集之前，必須先卸離或刪除主要叢集和次要叢集。

Note

此動作僅適用於 Amazon DocumentDB 叢集。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

GlobalClusterIdentifier

要刪除之全域叢集的叢集識別碼。

類型：字串

長度限制：長度下限為 1。長度上限為 255。

模式：`[A-Za-z][0-9A-Za-z-:._]*`

必要：是

回應元素

服務會傳回下列元素。

GlobalCluster

代表 Amazon 文件資料庫全域叢集的資料類型。

類型：[GlobalCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

GlobalClusterNotFoundFault

`GlobalClusterIdentifier` 不會參照現有的全域叢集。

HTTP 狀態碼：404

InvalidGlobalClusterStateFault

當群集處於此狀態時，請求的操作無法執行。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeCertificates

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回 Amazon DocumentDB 為此提供的憑證授權單位 (CA) 憑證清單。AWS 帳戶

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

CertificateIdentifier

使用者提供的憑證識別碼。如果指定此參數，則只會傳回指定憑證的資訊。如果省略此參數，則會傳回最多的MaxRecords憑證清單。此參數不區分大小寫。

限制

- 必須符合現有的CertificateIdentifier.

類型：字串

必要：否

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個 DescribeCertificates 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。若存在的記錄比 MaxRecords 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

預設：100

約束:

- 最小值：20
- 上限：100

類型：整數

必要：否

回應元素

服務會傳回下列元素。

證書. 證書 .N

此憑證的清單AWS 帳戶。

類型：[Certificate](#) 物件陣列

Marker

如果檢索到的記錄數大於，則提供可選的分頁令牌MaxRecords。如果指定了這個參數，標記會指定清單中的下一筆記錄。在下次呼叫Marker中包含的值，DescribeCertificates會產生憑證的下一頁。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

CertificateNotFound

CertificateIdentifier不參考現有憑證。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)

- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeDBClusterParameterGroups

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回 DBClusterParameterGroup 描述的清單。如果已指定 DBClusterParameterGroupName 參數，則清單只會包含指定叢集參數群組的描述。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterParameterGroupName

要傳回詳細資訊的特定叢集參數群組名稱。

約束：

- 如果提供，則必須與現有的名稱相符 DBClusterParameterGroup。

類型：字串

必要：否

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定 MaxRecords 值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

回應元素

服務會傳回下列元素。

數據ClusterParameterGroups庫 ClusterParameterGroup

叢集參數群組的清單。

類型：[DBClusterParameterGroup](#) 物件陣列

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBParameterGroupNotFound

DBParameterGroupName不參考現有的參數群組。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)

- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeDBClusterParameters

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回特定叢集參數群組的詳細參數清單。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterParameterGroupName

要傳回參數詳細資訊的特定叢集參數群組名稱。

約束：

- 如果提供，則必須與現有的名稱相符DBClusterParameterGroup。

類型：字串

必要：是

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定MaxRecords值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

Source

指出只傳回特定來源參數的值。參數來源可以是 engine、service 或 customer。

類型：字串

必要：否

回應元素

服務會傳回下列元素。

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

參數. 參數 .N

提供叢集參數群組的參數清單。

類型：[Parameter](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBParameterGroupNotFound

DBParameterGroupName 不參考現有的參數群組。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeDBClusters

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回已佈建的 Amazon DocumentDB 叢集的相關資訊。此 API 操作支持分頁。對於某些管理功能，例如叢集和執行個體生命週期管理，Amazon DocumentDB 會運用與 Amazon RDS 和 Amazon Neptune 共用的操作技術。使用 `filterName=engine,Values=docdb` 篩選器參數僅傳回 Amazon DocumentDB 叢集。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

使用者提供的叢集識別碼。如果指定此參數，則只會傳回來自特定叢集的資訊。此參數不區分大小寫。

約束：

- 如果提供，則必須與現有的相符 `DBClusterIdentifier`。

類型：字串

必要：否

篩選器. N

指定要描述之一或多個叢集的篩選器。

支援的篩選條件：

- `db-cluster-id`-接受叢集識別碼和叢集 Amazon 資源名稱 (ARN)。結果清單僅包含這些 ARN 所識別之叢集的相關資訊。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定MaxRecords值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

回應元素

服務會傳回下列元素。

資料庫叢集器 .n

叢集的清單。

類型：[DBCluster](#) 物件陣列

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到MaxRecords 指定的值為止。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

DBClusterIdentifier不參照現有的叢集。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeDBClusterSnapshotAttributes

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回手動資料庫叢集快照集的叢集快照屬性名稱和值的清單。

當您與其他快照共用時AWS 帳戶，會DescribeDBClusterSnapshotAttributes傳回已授權複製或還原手動叢集快照的restore屬性和 ID 清單。AWS 帳戶如果包含all在restore屬性的值清單中，則手動叢集快照會是公用的，所有人都可以複製或還原AWS 帳戶。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterSnapshotIdentifier

用來描述屬性的叢集快照識別碼。

類型：字串

必要：是

回應元素

服務會傳回下列元素。

DBClusterSnapshotAttributesResult

與叢集快照相關聯之屬性的詳細資訊。

類型：[DBClusterSnapshotAttributesResult](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier不會參考現有的叢集快照。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeDBClusterSnapshots

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回叢集快照的相關資訊。此 API 操作支持分頁。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

要擷取叢集快照清單的叢集識別碼。此參數無法與參數 `DBClusterSnapshotIdentifier` 數搭配使用。此參數不區分大小寫。

約束：

- 如果提供，則必須與現有的識別碼相符 `DBCluster`。

類型：字串

必要：否

DBClusterSnapshotIdentifier

要說明的特定叢集快照識別碼。此參數無法與參數 `DBClusterIdentifier` 數搭配使用。此值會以小寫字母字串的形式儲存。

約束：

- 如果提供，則必須與現有的識別碼相符 `DBClusterSnapshot`。
- 如果這是用於自動快照的識別符，也必須指定 `SnapshotType` 參數。

類型：字串

必要：否

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

IncludePublic

設定true為包含公用且可由任何人複製或還原的手動叢集快照AWS 帳戶，以及其他快照false。預設值為 false。

類型：布林值

必要：否

IncludeShared

設定true為包含其他AWS 帳戶已授予複製或還原權限AWS 帳戶的共用手動叢集快照，以及其他快照false。預設值為 false。

類型：布林值

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定MaxRecords值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

SnapshotType

要傳回的叢集快照類型。您可以指定下列其中一個值：

- automated-傳回 Amazon DocumentDB 為您自動建立的所有叢集快照。AWS 帳戶
- manual-傳回您為您手動建立的所有叢集快照AWS 帳戶。

- `shared`-傳回所有已共用給您的AWS 帳戶。
- `public`-傳回所有已標記為公用的叢集快照。

如果您未指定`SnapshotType`值，則會同時傳回自動和手動叢集快照。您可以將`IncludeShared`參數設定為`true`，以包含這些結果的共用叢集快照。您可以將`IncludePublic`參數設定為`true`，以包含這些結果的公用叢集快照。

`IncludeShared` 和 `IncludePublic` 參數不適用於值為 `SnapshotType` 的 `manual` 或 `automated`。當 `SnapshotType` 設為 `shared` 時，不適用 `IncludePublic` 參數。當 `SnapshotType` 設為 `public` 時，不適用 `IncludeShared` 參數。

類型：字串

必要：否

回應元素

服務會傳回下列元素。

數據ClusterSnapshots庫 ClusterSnapshot

提供叢集快照清單。

類型：[DBClusterSnapshot](#) 物件陣列

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`不會參考現有的叢集快照。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeDBEngineVersions

服務：Amazon DocumentDB (with MongoDB compatibility)

返回可用引擎的列表。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBParameterGroupFamily

要傳回詳細資料的特定參數群組族群的名稱。

約束：

- 如果提供，則必須與現有的相符DBParameterGroupFamily。

類型：字串

必要：否

DefaultOnly

指出只會傳回指定引擎或引擎預設版本和主要版本的組合。

類型：布林值

必要：否

Engine

要傳回的資料庫引擎。

類型：字串

必要：否

EngineVersion

要傳回的資料庫引擎版本。

範例：3.6.0

類型：字串

必要：否

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

ListSupportedCharacterSets

如已指定此參數，而請求的引擎支援 CreateDBInstance 的 CharacterSetName 參數，則回應會包含每個引擎版本支援的字元集清單。

類型：布林值

必要：否

ListSupportedTimezones

如已指定此參數，而請求的引擎支援 CreateDBInstance 的 TimeZone 參數，則回應會包含每個引擎版本支援的時區清單。

類型：布林值

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定 MaxRecords 值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

回應元素

服務會傳回下列元素。

數據EngineVersions庫 EngineVersion

—或多個引擎版本的詳細資訊。

類型：[DBEngineVersion](#) 物件陣列

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeDBInstances

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回佈建的 Amazon 文件資料庫執行個體的相關資訊。此 API 支援分頁。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBInstanceIdentifier

用戶提供的實例標識符。如果指定此參數，則只會傳回來自特定執行個體的資訊。此參數不區分大小寫。

約束：

- 如果提供，則必須與現有的識別碼相符DBInstance。

類型：字串

必要：否

篩選器. N

指定要描述之一或多個執行個體的篩選器。

支援的篩選條件：

- `db-cluster-id`-接受叢集識別碼和叢集 Amazon 資源名稱 (ARN)。結果清單僅包含與這些 ARN 識別之叢集相關聯之執行個體的相關資訊。
- `db-instance-id`-接受執行個體識別碼和執行個體 ARN。結果清單僅包含這些 ARN 所識別之實例的相關資訊。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定MaxRecords值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

回應元素

服務會傳回下列元素。

數據庫實例. N

一或多個執行個體的詳細資訊。

類型：[DBInstance](#) 物件陣列

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到MaxRecords 指定的值為止。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBInstanceNotFound

DBInstanceIdentifier不引用現有實例。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeDBSubnetGroups

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回 DBSubnetGroup 描述的清單。如果指定 DBSubnetGroupName 了，清單將僅包含指定的描述 DBSubnetGroup。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBSubnetGroupName

要傳回詳細資訊的子網路群組名稱。

類型：字串

必要：否

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定 MaxRecords 值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

回應元素

服務會傳回下列元素。

數據SubnetGroups庫 SubnetGroup

有關一或多個子網路群組的詳細資訊。

類型：[DBSubnetGroup](#) 物件陣列

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBSubnetGroupNotFoundFault

DBSubnetGroupName不參照現有的子網路群組。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)

- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeEngineDefaultClusterParameters

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回叢集資料庫引擎的預設引擎和系統參數資訊。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBParameterGroupFamily

要傳回引擎參數資訊的叢集參數群組族群名稱。

類型：字串

必要：是

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定MaxRecords值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

回應元素

服務會傳回下列元素。

EngineDefaults

包含成功呼叫DescribeEngineDefaultClusterParameters作業的結果。

類型：[EngineDefaults](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeEventCategories

服務：Amazon DocumentDB (with MongoDB compatibility)

顯示所有事件來源類型或特定來源類型 (如果指定) 的類別清單。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

SourceType

產生事件的來源類型。

有效值：db-instance、db-parameter-group、db-security-group

類型：字串

必要：否

回應元素

服務會傳回下列元素。

EventCategoriesMapList。 EventCategoriesMap.N

事件類別地圖的列表。

類型：[EventCategoriesMap](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeEvents

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回過去 14 天內與執行個體、安全群組、快照和資料庫參數群組相關的事件。您可以透過提供名稱作為參數，以取得特定資料庫執行個體、安全群組、快照或參數群組特有的事件。依預設，會傳回過去一小時的事件。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

Duration

擷取事件的分鐘數。

預設：60

類型：整數

必要：否

EndTime

擷取事件的時間間隔終點，以 ISO 8601 格式指定。

範例：2009-07-08T18:00Z

類型：Timestamp

必要：否

EventCategories。 EventCategory.N

觸發事件通知訂閱通知的事件類別清單。

類型：字串陣列

必要：否

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定 MaxRecords 值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

SourceIdentifier

要傳回其事件的事件來源識別碼。如未指定，回應中會包含所有來源。

約束：

- 如果 SourceIdentifier 有提供，也 SourceType 必須提供。
- 如果來源類型為 DBInstance，則 DBInstanceIdentifier 必須提供。
- 如果來源類型為 DBSecurityGroup，則 DBSecurityGroupName 必須提供。
- 如果來源類型為 DBParameterGroup，則 DBParameterGroupName 必須提供。
- 如果來源類型為 DBSnapshot，則 DBSnapshotIdentifier 必須提供。
- 不能以連字號結尾或連續包含兩個連字號。

類型：字串

必要：否

SourceType

事件擷取來源的事件來源。如未指定任何值，則會傳回所有事件。

類型：字串

有效值:db-instance | db-parameter-group | db-security-group | db-snapshot
| db-cluster | db-cluster-snapshot

必要：否

StartTime

擷取事件的時間間隔起點，以 ISO 8601 格式指定。

範例：2009-07-08T18:00Z

類型：Timestamp

必要：否

回應元素

服務會傳回下列元素。

活動事件 N

有關一個或多個事件的詳細信息。

類型：[Event](#) 物件陣列

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)

- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeEventSubscriptions

服務：Amazon DocumentDB (with MongoDB compatibility)

列出客戶帳戶的所有訂閱描述。訂閱的說明包

括SubscriptionNameSNSTopicARNCustomerID、SourceType、SourceID、CreationTime、和Status。

如果您指定SubscriptionName，會列出該訂閱的說明。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定MaxRecords值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

SubscriptionName

您要描述的 Amazon DocumentDB 事件通知訂閱的名稱。

類型：字串

必要：否

回應元素

服務會傳回下列元素。

EventSubscriptionsList。 EventSubscription.N

事件訂閱清單。

類型：[EventSubscription](#) 物件陣列

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

SubscriptionNotFound

訂閱名稱不存在。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)

- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeGlobalClusters

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回有關 Amazon DocumentDB 全域叢集的資訊。此 API 支援分頁。

Note

此動作僅適用於 Amazon DocumentDB 叢集。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

篩選器. N

指定一或多個要描述的全域資料庫叢集的篩選器。

支援的篩選器：`db-cluster-id`接受叢集識別碼和叢集 Amazon 資源名稱 (ARN)。結果清單只會包含這些 ARN 所識別之叢集的相關資訊。

類型：[Filter](#) 物件陣列

必要：否

GlobalClusterIdentifier

使用者提供的叢集識別碼。如果指定此參數，則只會傳回來自特定叢集的資訊。此參數沒有大小寫之分。

類型：字串

長度限制：長度下限為 1。長度上限為 255。

模式：`[A-Za-z][0-9A-Za-z-:._]*`

必要：否

Marker

前一個 `DescribeGlobalClusters` 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄超過指定MaxRecords值，則響應中包含稱為標記的分頁令牌，以便您可以檢索剩餘的結果。

類型：整數

必要：否

回應元素

服務會傳回下列元素。

GlobalClusters。 GlobalClusterMember.N

類型：[GlobalCluster](#) 物件陣列

Marker

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

GlobalClusterNotFoundFault

GlobalClusterIdentifier不會參照現有的全域叢集。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)

- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribeOrderableDBInstanceOptions

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回指定引擎的可訂購執行個體選項清單。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

Engine

要擷取執行個體選項的引擎名稱。

類型：字串

必要：是

DBInstanceClass

實體類別篩選器值。指定此參數，只會顯示符合指定執行處理類別的可用提供項目。

類型：字串

必要：否

EngineVersion

引擎版本篩選條件值。指定此參數以僅顯示符合指定引擎版本的可用供應項目。

類型：字串

必要：否

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

LicenseModel

授權模型篩選條件值。指定此參數以僅顯示符合指定授權模式的可用供應項目。

類型：字串

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定 MaxRecords 值，則響應中包含分頁令牌（標記），以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

Vpc

虛擬私有雲 (VPC) 篩選器值。指定此參數，以僅顯示可用的 VPC 或非 VPC 供應項目。

類型：布林值

必要：否

回應元素

服務會傳回下列元素。

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

可訂購數據庫 . InstanceOptions 可訂購數據庫 .N InstanceOption

特定可訂購執行處理可用的選項。

類型：[OrderableDBInstanceOption](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

DescribePendingMaintenanceActions

服務：Amazon DocumentDB (with MongoDB compatibility)

傳回至少具有一個擱置維護動作的資源清單 (例如執行個體)。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

篩選器. N

篩選條件，指定傳回待處理維護動作的一或多種資源。

支援的篩選條件：

- `db-cluster-id`-接受叢集識別碼和叢集 Amazon 資源名稱 (ARN)。結果清單僅包含這些 ARN 識別之叢集的擱置維護動作。
- `db-instance-id`-接受執行個體識別碼和執行個體 ARN。結果清單僅包含這些 ARN 識別之資料庫執行個體的擱置維護動作。

類型：[Filter](#) 物件陣列

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

類型：字串

必要：否

MaxRecords

要在回應中包含的記錄數量上限。如果存在的記錄多於指定 `MaxRecords` 值，則響應中包含分頁令牌 (標記)，以便可以檢索剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

類型：整數

必要：否

ResourceIdentifier

傳回待處理維護動作的資源 ARN。

類型：字串

必要：否

回應元素

服務會傳回下列元素。

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

類型：字串

PendingMaintenanceActions。 ResourcePendingMaintenanceActions.N

要套用的維護動作。

類型：[ResourcePendingMaintenanceActions](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ResourceNotFoundFault

找不到指定的資源 ID。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

FailoverDBCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

強制叢集的容錯移轉。

叢集的容錯移轉會將叢集中的其中一個 Amazon DocumentDB 複本 (唯讀執行個體) 提升為主要執行個體 (叢集寫入器)。

如果主執行個體發生故障，Amazon DocumentDB 會自動容錯移轉到 Amazon DocumentDB 複本 (如果有的話)。當您想要模擬主要執行個體故障以進行測試時，可以強制容錯移轉。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

用於強制容錯移轉的叢集識別碼。此參數不區分大小寫。

約束：

- 必須符合現有 DBCluster 的識別碼。

類型：字串

必要：否

TargetDBInstanceIdentifier

要提升為主要執行個體的執行個體名稱。

您必須為叢集中的 Amazon DocumentDB 複本指定執行個體識別碼。例如 mydbcluster-replica1。

類型：字串

必要：否

回應元素

服務會傳回下列元素。

DBCluster

叢集的詳細資訊。

類型：[DBCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

`DBClusterIdentifier`不參照現有的叢集。

HTTP 狀態碼：404

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ListTagsForResource

服務：Amazon DocumentDB (with MongoDB compatibility)

列出 Amazon DocumentDB 資源上的所有標籤。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

ResourceName

與要列出的標籤 Amazon DocumentDB 資源。這個值是 Amazon Resource Name (ARN)。

類型：字串

必要：是

篩選器. N

目前不支援此參數。

類型：[Filter](#) 物件陣列

必要：否

回應元素

服務會傳回下列元素。

TagList. 標籤

一或多個標籤的清單。

類型：[Tag](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

DBClusterNotFoundFault

`DBClusterIdentifier` 不參考現有叢集。

HTTP 狀態碼：404

DBInstanceNotFound

DBInstanceIdentifier 不引用現有實例。

HTTP 狀態碼：404

DBSnapshotNotFound

DBSnapshotIdentifier 不參考現有的快照。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ModifyDBCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

修改 Amazon DocumentDB 叢集的設定。您可以透過在請求中指定這些參數及新的值，來變更一或多個資料庫組態參數。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

要修改之叢集的叢集識別碼。此參數不區分大小寫。

約束：

- 必須符合現有 DBCluster 的識別碼。

類型：字串

必要：是

AllowMajorVersionUpgrade

指示是否允許升級主要版本的值。

限制：當指定與資料庫叢集目前版本不同之主要版本的EngineVersion參數值時，您必須允許主要版本升級。

類型：布林值

必要：否

ApplyImmediately

指定是否會儘快非同步套用此要求中的變更和任何暫止變更的值，而不論叢集的PreferredMaintenanceWindow設定為何。如果將此參數設定為false，則會在下一個維護時段套用對叢集的變更。

ApplyImmediately參數只會影響NewDBClusterIdentifier和MasterUserPassword值。如果將此參數值設定為false，則會在下一個維護時段套用對NewDBClusterIdentifier和MasterUserPassword值的變更。所有其他的變更都會立即套用，無論 ApplyImmediately 參數的值為何。

預設：false

類型：布林值

必要：否

BackupRetentionPeriod

自動備份保留的天數。您必須指定一個值 (最小值為 1)。

預設：1

約束：

- 該值必須介於 1 到 35 之間。

類型：整數

必要：否

CloudwatchLogsExportConfiguration

針對特定執行個體或叢集啟用匯出至 Amazon CloudWatch Logs 的日誌類型的組態設定。EnableLogTypes和DisableLogTypes陣列會決定哪些記錄要匯出 (或不匯出) 至 CloudWatch 記錄。

類型：[CloudwatchLogsExportConfiguration](#) 物件

必要：否

DBClusterParameterGroupName

要用於叢集的叢集參數群組名稱。

類型：字串

必要：否

DeletionProtection

指定是否可以刪除此叢集。如果啟DeletionProtection用，除非已修改並DeletionProtection停用叢集，否則無法刪除叢集。DeletionProtection防止叢集遭到意外刪除。

類型：布林值

必要：否

EngineVersion

您要升級的資料庫引擎版本號碼。變更此參數會造成中斷。變更會在下一次維護時段期間套用，除非已啟用 ApplyImmediately。

若要列出 Amazon DocumentDB 的所有可用引擎版本，請使用下列命令：

```
aws docdb describe-db-engine-versions --engine docdb --query
"DBEngineVersions[].EngineVersion"
```

類型：字串

必要：否

MasterUserPassword

主要資料庫使用者的密碼。此密碼可包含任何可印出的 ASCII 字元，但斜線 (/)、雙引號 (") 或 "at" 符號 (@) 則除外。

限制條件：必須包含 8 到 100 個字元。

類型：字串

必要：否

NewDBClusterIdentifier

重新命名叢集時，叢集的新叢集識別碼。此值會以小寫字母字串的形式儲存。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：my-cluster2

類型：字串

必要：否

Port

叢集接受連線的連接埠號碼。

限制：必須是從1150到的值65535。

預設值：與原始叢集相同的連接埠。

類型：整數

必要：否

PreferredBackupWindow

每日的時間範圍，若使用 BackupRetentionPeriod 參數啟用自動化備份，則會在此期間建立自動化備份。

預設是從各 AWS 區域 8 小時時段內隨機選取的 30 分鐘時段。

約束：

- 格式必須為 hh24:mi-hh24:mi。
- 必須以國際標準時間 (UTC) 表示。
- 不得和慣用的維護時段衝突。
- 必須至少 30 分鐘。

類型：字串

必要：否

PreferredMaintenanceWindow

每週可能進行系統維護的時段，以國際標準時間 (UTC) 表示。

格式：ddd:hh24:mi-ddd:hh24:mi

預設是從各 AWS 區域 8 小時時段內隨機選取的 30 分鐘時段，並隨機發生在一週內的某一天。

有效日：星期一、星期二、星期三、星期四、星期五、星期六、星期日

限制條件：必須至少是 30 分鐘的時段。

類型：字串

必要：否

StorageType

和資料庫叢集相關聯的儲存類型。

如需 Amazon DocumentDB 叢集的儲存類型的相關資訊，請參閱 Amazon DocumentDB 開發人員指南中的叢集儲存組態。

儲存類型的有效值-standard | iopt1

預設值為 standard

類型：字串

必要：否

VpcSecurityGroupIds。 VpcSecurityGroupId.N

叢集將屬於的虛擬私人雲端 (VPC) 安全性群組清單。

類型：字串陣列

必要：否

回應元素

服務會傳回下列元素。

DBCluster

叢集的詳細資訊。

類型：[DBCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterAlreadyExistsFault

您已經擁有具有指定識別碼的叢集。

HTTP 狀態碼：400

DBClusterNotFoundFault

DBClusterIdentifier不參考現有叢集。

HTTP 狀態碼：404

DBClusterParameterGroupNotFound

DBClusterParameterGroupName不參考現有的叢集參數群組。

HTTP 狀態碼：404

DBSubnetGroupNotFoundFault

DBSubnetGroupName不參照現有的子網路群組。

HTTP 狀態碼：404

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

InvalidDBSecurityGroupState

安全性群組的狀態不允許刪除。

HTTP 狀態碼：400

InvalidDBSubnetGroupStateFault

子網路群組無法刪除，因為它正在使用中。

HTTP 狀態碼：400

InvalidSubnet

要求的子網路無效，或要求多個子網路並非全部位於一般虛擬私人雲端 (VPC) 中。

HTTP 狀態碼：400

InvalidVPCNetworkStateFault

子網路群組在建立之後，並不會涵蓋所有可用區域，因為已進行變更。

HTTP 狀態碼：400

StorageQuotaExceeded

該請求會導致您超過所有執行個體允許的可用儲存空間量。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ModifyDBClusterParameterGroup

服務：Amazon DocumentDB (with MongoDB compatibility)

修改叢集參數群組的參數。若要修改超過一個參數，請提交下列項目的清

單：ParameterName、ParameterValue 和 ApplyMethod。單一請求中最多可修改 20 個參數。

Note

動態參數的變更會立即套用。變更靜態參數需要重新開機或維護時段，變更才能生效。

Important

建立叢集參數群組後，您應等待至少 5 分鐘，然後再建立第一個使用該叢集參數群組做為預設參數群組的叢集。這可讓 Amazon DocumentDB 在將參數群組用作新叢集的預設值之前，完全完成建立動作。此步驟對建立叢集預設資料庫時的關鍵參數尤其重要，像是 `character_set_database` 參數所定義的預設資料庫字元集。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBClusterParameterGroupName

要修改之叢集參數群組的名稱。

類型：字串

必要：是

參數. 參數 .N

叢集參數群組中要修改的參數清單。

類型：[Parameter](#) 物件陣列

必要：是

回應元素

服務會傳回下列元素。

DBClusterParameterGroupName

叢集參數群組的名稱。

約束:

- 必須介於 1 至 255 個字母或數字之間。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

Note

此值會以小寫字母字串的形式儲存。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBParameterGroupNotFound

DBParameterGroupName 不參考現有的參數群組。

HTTP 狀態碼：404

InvalidDBParameterGroupState

參數群組正在使用中，或處於無效狀態。如果您嘗試刪除參數群組，則無法在參數群組處於此狀態時將其刪除。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)

- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ModifyDBClusterSnapshotAttribute

服務：Amazon DocumentDB (with MongoDB compatibility)

將屬性和值新增至手動叢集快照，或從中移除屬性和值。

若要與其他人共用手動叢集快照 AWS 帳戶 `AttributeName`，請指定 `restore` 為，然後使用 `ValuesToAdd` 參數新增授權還原手動叢集快照的 ID 清單。AWS 帳戶使用此值可將手動叢集快照設 `all` 為公用，這表示所有人都可以複製或還原該快照 AWS 帳戶。請勿 `all` 為包含您不希望所有人使用的私人資訊的任何手動叢集快照新增此值 AWS 帳戶。如果手動叢集快照已加密，則可以共用該快照，但只能指定 `ValuesToAdd` 參數的授權 AWS 帳戶 ID 清單。本例中無法使用 `all` 做為該參數的值。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

AttributeName

要修改的叢集快照屬性名稱。

若要管理其他 AWS 帳戶人複製或還原手動叢集快照的授權，請將此值設定為 `restore`。

類型：字串

必要：是

DBClusterSnapshotIdentifier

要修改其屬性的叢集快照識別碼。

類型：字串

必要：是

ValuesToAdd。 AttributeValue.N

要新增至指定之屬性的叢集快照屬性清單 `AttributeName`。

若要授權其他 AWS 帳戶人複製或還原手動叢集快照，請將此清單設定為包含一或多個 AWS 帳戶 ID。若要讓手動叢集快照可由任何人還原 AWS 帳戶，請將其設定為 `all`。請勿 `all` 為任何包含您不想提供給所有人使用的私人資訊的手動叢集快照新增此值 AWS 帳戶。

類型：字串陣列

必要：否

ValuesToRemove。 AttributeValue.N

要從指定的屬性中移除的叢集快照屬性清單AttributeName。

若要移除其他AWS 帳戶人複製或還原手動叢集快照的授權，請將此清單設定為包含一或多個AWS 帳戶識別碼。若要移除任AWS 帳戶何複製或還原叢集快照的授權，請將其設定為all。如果您指定all，AWS 帳戶其帳戶 ID 已明確新增至restore屬性，仍然可以複製或還原手動叢集快照。

類型：字串陣列

必要：否

回應元素

服務會傳回下列元素。

DBClusterSnapshotAttributesResult

與叢集快照相關聯之屬性的詳細資訊。

類型：[DBClusterSnapshotAttributesResult](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier不會參考現有的叢集快照。

HTTP 狀態碼：404

InvalidDBClusterSnapshotStateFault

提供的值不是有效的叢集快照狀態。

HTTP 狀態碼：400

SharedSnapshotQuotaExceeded

您已超過可共享手動資料庫快照的帳戶數量上限。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ModifyDBInstance

服務：Amazon DocumentDB (with MongoDB compatibility)

修改執行個體的設定。您可以透過在請求中指定這些參數及新的值，來變更一或多個資料庫組態參數。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBInstanceIdentifier

執行個體識別碼。此值會以小寫字母字串的形式儲存。

約束：

- 必須符合現有 DBInstance 的識別碼。

類型：字串

必要：是

ApplyImmediately

指定是否會儘快以非同步方式套用此要求中的修改和任何擱置的修改，而不論執行個體的 PreferredMaintenanceWindow 設定為何。

如果將此參數設定為 false，則會在下一個維護時段套用對例證的變更。某些參數變更可能會導致中斷，並在下次重新開機時套用。

預設：false

類型：布林值

必要：否

AutoMinorVersionUpgrade

此參數不適用於 Amazon DocumentDB。無論設定的值為何，Amazon DocumentDB 都不會執行次要版本升級。

類型：布林值

必要：否

CACertificateIdentifier

指出需要與執行個體建立關聯的憑證。

類型：字串

必要：否

CertificateRotationRestart

指定當您輪換 SSL/TLS 憑證時是否重新啟動資料庫執行個體。

根據預設，當您輪換 SSL/TLS 憑證時，資料庫執行個體會重新啟動。在資料庫執行個體重新啟動之前，憑證不會更新。

Important

只有在未使用 SSL/TLS 連線至資料庫執行個體時，才設定此參數。

如果您使用 SSL/TLS 連線到資料庫執行個體，請參閱 [Amazon DocumentDB 開發人員指南中的更新您的 Amazon 文件資料庫 TLS 憑證和加密傳輸中的資料](#)。

類型：布林值

必要：否

CopyTagsToSnapshot

指出是否要將資料庫執行個體的所有標籤複製到資料庫執行個體快照的值。根據預設，不會複製標籤。

類型：布林值

必要：否

DBInstanceClass

執行個體的新運算和記憶體容量；例如，`db.r5.large`。並非所有實例類都可用AWS 區域。

如果您修改執行個體類別，變更期間就會發生中斷。變更會在下一個維護時段期間套用，除非針對此請求將 `ApplyImmediately` 指定為 `true`。

預設值：使用現有設定。

類型：字串

必要：否

EnablePerformanceInsights

指示是否啟用資料庫執行個體績效詳情的值。如需詳細資訊，請參閱[使用 Amazon 績效詳情](#)。

類型：布林值

必要：否

NewDBInstanceIdentifier

重新命名執行個體時，執行個體的新執行個體識別碼。當您變更執行個體識別碼時，如果您設定Apply Immediately為，則會立即重新啟動執行個體true。如果您設定Apply Immediately為，則會在下一個維護時段中發生false。此值會以小寫字母字串的形式儲存。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：mydbinstance

類型：字串

必要：否

PerformanceInsightsKMSKeyId

用於 Performance Insights 資料加密的 AWS KMS 金鑰識別符。

金AWS KMS鑰識別碼是 KMS 金鑰的金鑰 ARN、金鑰識別碼、別名 ARN 或別名。

如果您沒有為 PerformanceInsights KMS 指定值KeyId，則 Amazon DocumentDB 會使用您的預設 KMS 金鑰。您的 Amazon Web Services 帳戶有一個默認的 KMS 密鑰。您的 Amazon Web Services 帳戶對每個亞 Amazon Web Services 區域都有不同的預設 KMS 金鑰。

類型：字串

必要：否

PreferredMaintenanceWindow

每週可能進行系統維護的時間範圍 (UTC)，在此期間可能會發生中斷。除了下列情況外，變更此參數不會導致中斷，而且會儘快以非同步方式套用變更。如果存在導致重新開機的擱置動作，且維護

時段已變更為包含目前的時間，變更此參數會導致執行個體重新開機。如果您要將此視窗移到目前時間，則在目前時間和視窗結束之間至少必須有 30 分鐘，以確保套用暫止的變更。

預設值：使用現有設定。

格式：ddd:hh24:mi-ddd:hh24:mi

有效日：星期一、星期二、星期三、星期四、星期五、星期六、星期日

限制：必須至少為 30 分鐘。

類型：字串

必要：否

PromotionTier

一個值，指定 Amazon DocumentDB 複本在現有主執行個體發生故障後提升為主執行個體的順序。

預設：1

有效值：0 至 15

類型：整數

必要：否

回應元素

服務會傳回下列元素。

DBInstance

執行個體的詳細資訊。

類型：[DBInstance](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AuthorizationNotFound

指定的 CIDR IP 或 Amazon EC2 安全群組未獲得指定安全群組的授權。

Amazon DocumentDB 也可能無法獲得使用 IAM 代表您執行必要動作的授權。

HTTP 狀態碼：404

CertificateNotFound

CertificateIdentifier 不參考現有憑證。

HTTP 狀態碼：404

DBInstanceAlreadyExists

您已經擁有具有指定識別碼的執行個體。

HTTP 狀態碼：400

DBInstanceNotFound

DBInstanceIdentifier 不會參考現有的執行個體。

HTTP 狀態碼：404

DBParameterGroupNotFound

DBParameterGroupName 不參考現有的參數群組。

HTTP 狀態碼：404

DBSecurityGroupNotFound

DBSecurityGroupName 不參考現有的安全性群組。

HTTP 狀態碼：404

DBUpgradeDependencyFailure

升級失敗，因為無法修改相依的資源。

HTTP 狀態碼：400

InsufficientDBInstanceCapacity

指定的執行個體類別在指定的可用區域中無法使用。

HTTP 狀態碼：400

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

InvalidDBSecurityGroupState

安全性群組的狀態不允許刪除。

HTTP 狀態碼：400

InvalidVPCNetworkStateFault

子網路群組在建立之後，並不會涵蓋所有可用區域，因為已進行變更。

HTTP 狀態碼：400

StorageQuotaExceeded

該請求將導致您超過所有實例允許的可用存儲量。

HTTP 狀態碼：400

StorageTypeNotSupported

指定的儲存體StorageType無法與資料庫執行個體相關聯。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ModifyDBSubnetGroup

服務：Amazon DocumentDB (with MongoDB compatibility)

修改現有的子網路群組。子網路群組必須至少包含一個子網路，位於中的兩個可用區域。AWS 區域

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBSubnetGroupName

子網路群組的名稱。此值會以小寫字母字串的形式儲存。您無法修改預設子網路群組。

限制條件：必須符合現有 DBSubnetGroup 的名稱。絕不能為預設值。

範例：mySubnetgroup

類型：字串

必要：是

SubnetIds。 SubnetIdentifier.N

子網路群組的 Amazon EC2 子網路 ID。

類型：字串陣列

必要：是

DBSubnetGroupDescription

子網路群組的描述。

類型：字串

必要：否

回應元素

服務會傳回下列元素。

DBSubnetGroup

子網路群組的詳細資訊。

類型：[DBSubnetGroup](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBSubnetGroupDoesNotCoverEnoughAZs

除非只有一個可用區域，否則子網路群組中的子網路應至少涵蓋兩個可用區域。

HTTP 狀態碼：400

DBSubnetGroupNotFoundFault

DBSubnetGroupName 不參照現有的子網路群組。

HTTP 狀態碼：404

DBSubnetQuotaExceededFault

此要求會導致您超過子網路群組中允許的子網路數目。

HTTP 狀態碼：400

InvalidSubnet

要求的子網路無效，或要求多個子網路並非全部位於一般虛擬私人雲端 (VPC) 中。

HTTP 狀態碼：400

SubnetAlreadyInUse

子網路已在可用區域中使用。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)

- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ModifyEventSubscription

服務：Amazon DocumentDB (with MongoDB compatibility)

修改現有的 Amazon DocumentDB 事件通知訂閱。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

SubscriptionName

Amazon DocumentDB 事件通知訂閱的名稱。

類型：字串

必要：是

Enabled

布林值；設定true為啟動訂閱。

類型：布林值

必要：否

EventCategories。 EventCategory.N

您要訂閱的事件類別清單。SourceType

類型：字串陣列

必要：否

SnsTopicArn

針對事件通知所建立 SNS 主題的 Amazon Resource Name (ARN)。此 ARN 是當您建立並訂閱主題時由 Amazon SNS 所建立。

類型：字串

必要：否

SourceType

產生事件的來源類型。例如，如果您想要收到執行處理所產生的事件通知，請將此參數設定為db-instance。若沒有指定此值，則會傳回所有事件。

有效值：db-instance、db-parameter-group、db-security-group

類型：字串

必要：否

回應元素

服務會傳回下列元素。

EventSubscription

有關您已訂閱之事件的詳細資訊。

類型：[EventSubscription](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

EventSubscriptionQuotaExceeded

你已達到活動訂閱數目上限。

HTTP 狀態碼：400

SNSInvalidTopic

Amazon SNS 已回應指定主題存在問題。

HTTP 狀態碼：400

SNSNoAuthorization

您沒有發佈至 SNS 主題 Amazon 資源名稱 (ARN) 的權限。

HTTP 狀態碼：400

SNSTopicArnNotFound

SNS 主題 Amazon 資源名稱 (ARN) 不存在。

HTTP 狀態碼：404

SubscriptionCategoryNotFound

提供的類別不存在。

HTTP 狀態碼：404

SubscriptionNotFound

訂閱名稱不存在。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ModifyGlobalCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

修改 Amazon DocumentDB 全域叢集的設定。您可以在要求中指定這些參數和新值，來變更一或多個組態參數 (例如：刪除保護) 或全域叢集識別碼。

Note

此動作僅適用於 Amazon DocumentDB 叢集。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

GlobalClusterIdentifier

要修改之全域叢集的識別碼。此參數沒有大小寫之分。

約束：

- 必須符合現有全域叢集的識別碼。

類型：字串

長度限制：長度下限為 1。長度上限為 255。

模式：[A-Za-z][0-9A-Za-z-:._]*

必要：是

DeletionProtection

指出全域叢集是否已啟用刪除保護。啟用刪除保護時，無法刪除全域叢集。

類型：布林值

必要：否

NewGlobalClusterIdentifier

當您修改全域叢集時，全域叢集的新識別碼。此值會以小寫字母字串的形式儲存。

- 必須包含 1 到 63 個字母、數字或連字號

第一個字元必須是字母

不能以一個連字號結尾或是連續包含兩個連字號

範例：my-cluster2

類型：字串

長度限制：長度下限為 1。長度上限為 255。

模式：[A-Za-z][0-9A-Za-z-:._]*

必要：否

回應元素

服務會傳回下列元素。

GlobalCluster

代表 Amazon 文件資料庫全域叢集的資料類型。

類型：[GlobalCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

GlobalClusterNotFoundFault

GlobalClusterIdentifier 不會參照現有的全域叢集。

HTTP 狀態碼：404

InvalidGlobalClusterStateFault

當群集處於此狀態時，請求的操作無法執行。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

RebootDBInstance

服務：Amazon DocumentDB (with MongoDB compatibility)

您可能需要重新啟動執行個體，通常是出於維護原因。例如，如果您進行某些變更，或變更與執行個體相關聯的叢集參數群組，則必須重新啟動執行個體，變更才會生效。

重新啟動執行個體，將重新啟動資料庫引擎服務。重新啟動執行個體會導致暫時中斷，在此期間，執行個體狀態會設定為重新啟動。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBInstanceIdentifier

執行個體識別碼。此參數是以小寫字母字串的形式儲存。

約束：

- 必須符合現有 DBInstance 的識別碼。

類型：字串

必要：是

ForceFailover

何時 true，重新開機是透過異地同步備份容錯移轉來執行。

限制：您無法指定執行個體是 true 否未針對異地同步備份設定。

類型：布林值

必要：否

回應元素

服務會傳回下列元素。

DBInstance

執行個體的詳細資訊。

類型：[DBInstance](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBInstanceNotFound

`DBInstanceIdentifier`不會參考現有的執行個體。

HTTP 狀態碼：404

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

RemoveFromGlobalCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

從全域叢集中卸離 Amazon DocumentDB 次要叢集。叢集會變成具有讀寫功能的獨立叢集，而不是唯讀叢集，並從不同區域的主要叢集接收資料。

Note

此動作僅適用於 Amazon DocumentDB 叢集。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DbClusterIdentifier

Amazon 資源名稱 (ARN) 可識別從 Amazon DocumentDB 全域叢集中分離的叢集。

類型：字串

必要：是

GlobalClusterIdentifier

要從 Amazon DocumentDB 全域叢集中分離的叢集識別碼。

類型：字串

長度限制：長度下限為 1。長度上限為 255。

模式：[A-Za-z][0-9A-Za-z-:._]*

必要：是

回應元素

服務會傳回下列元素。

GlobalCluster

代表 Amazon 文件資料庫全域叢集的資料類型。

類型：[GlobalCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

`DBClusterIdentifier`不參照現有的叢集。

HTTP 狀態碼：404

GlobalClusterNotFoundFault

`GlobalClusterIdentifier`不會參照現有的全域叢集。

HTTP 狀態碼：404

InvalidGlobalClusterStateFault

當群集處於此狀態時，請求的操作無法執行。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

RemoveSourceIdentifierFromSubscription

服務：Amazon DocumentDB (with MongoDB compatibility)

從現有的 Amazon 文件資料庫事件通知訂閱中移除來源識別碼。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

SourceIdentifier

要從訂閱移除的來源識別碼，例如執行個體的執行個體識別碼，或安全性群組的名稱。

類型：字串

必要：是

SubscriptionName

您要從中移除來源識別碼之 Amazon DocumentDB 事件通知訂閱的名稱。

類型：字串

必要：是

回應元素

服務會傳回下列元素。

EventSubscription

有關您已訂閱之事件的詳細資訊。

類型：[EventSubscription](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

SourceNotFound

找不到要求的來源。

HTTP 狀態碼：404

SubscriptionNotFound

訂閱名稱不存在。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

RemoveTagsFromResource

服務：Amazon DocumentDB (with MongoDB compatibility)

從 Amazon DocumentDB 資源中刪除元數據標籤。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

ResourceName

該標籤被刪除 Amazon DocumentDB 資源。這個值是 Amazon Resource Name (ARN)。

類型：字串

必要：是

TagKeys. 成員.

要移除之標籤的標籤金鑰 (名稱)。

類型：字串陣列

必要：是

錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

DBClusterNotFoundFault

`DBClusterIdentifier` 不參考現有叢集。

HTTP 狀態碼：404

DBInstanceNotFound

`DBInstanceIdentifier` 不引用現有實例。

HTTP 狀態碼：404

DBSnapshotNotFound

`DBSnapshotIdentifier` 不參考現有的快照。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

ResetDBClusterParameterGroup

服務：Amazon DocumentDB (with MongoDB compatibility)

將叢集參數群組的參數修改為預設值。若要重設特定參數，請提交下列清單：ParameterName和ApplyMethod。若要重設整個叢集參數群組，請指定DBClusterParameterGroupName和ResetAllParameters參數。

重設整個群組時，動態參數會立即更新，靜態參數設定為pending-reboot在下次資料庫執行個體重啟時生效。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBClusterParameterGroupName

要重設之叢集參數群組的名稱。

類型：字串

必要：是

參數. 參數 .N

叢集參數群組中要重設為預設值的參數名稱清單。若將 ResetAllParameters 參數設為 true，您便無法使用此參數。

類型：[Parameter](#) 物件陣列

必要：否

ResetAllParameters

設定為true將叢集參數群組中所有參數重設為其預設值的值，false否則會設定為。若有為 Parameters 參數指定參數名稱清單，您便無法使用此參數。

類型：布林值

必要：否

回應元素

服務會傳回下列元素。

DBClusterParameterGroupName

叢集參數群組的名稱。

約束：

- 必須介於 1 至 255 個字母或數字之間。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

Note

此值會以小寫字母字串的形式儲存。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBParameterGroupNotFound

DBParameterGroupName 不參考現有的參數群組。

HTTP 狀態碼：404

InvalidDBParameterGroupState

參數群組正在使用中，或處於無效狀態。如果您嘗試刪除參數群組，則無法在參數群組處於此狀態時將其刪除。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)

- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

RestoreDBClusterFromSnapshot

服務：Amazon DocumentDB (with MongoDB compatibility)

從快照或叢集快照建立新叢集。

如果指定快照，則會使用預設組態和預設安全性群組從來源資料庫快照建立目標叢集。

如果指定叢集快照，則會使用與原始來源資料庫叢集相同的組態，從來源叢集還原點建立目標叢集，但新叢集是使用預設安全性群組建立的。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

要從快照或叢集快照建立的叢集名稱。此參數不區分大小寫。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：my-snapshot-id

類型：字串

必要：是

Engine

用於新叢集的資料庫引擎。

預設值：與來源相同。

約束：必須與來源的引擎相容。

類型：字串

必要：是

SnapshotIdentifier

還原來源的快照或叢集快照識別碼。

您可以使用名稱或 Amazon Resource Name (ARN) 來指定叢集快照。不過，您可以只使用 ARN 來指定快照。

約束:

- 必須符合現有快照的識別碼。

類型：字串

必要：是

AvailabilityZones。 AvailabilityZone.N

提供可在其中建立還原資料庫叢集中執行個體的 Amazon EC2 可用區域清單。

類型：字串陣列

必要：否

DBClusterParameterGroupName

要與此資料庫叢集建立關聯的資料庫叢集參數群組名稱。

類型：字串 需要：否。

如果省略此引數，則會使用預設的資料庫叢集參數群組。如果提供，則必須符合現有預設資料庫叢集參數群組的名稱。字串必須由 1 至 255 個字母、數字或連字號組成。它的第一個字符必須是一個字母，並且不能以連字符結束或包含兩個連續的連字符。

類型：字串

必要：否

DBSubnetGroupName

要用於新叢集的子網路群組名稱。

條件約束：如果提供，則必須與現有的名稱相符DBSubnetGroup。

範例：mySubnetgroup

類型：字串

必要：否

DeletionProtection

指定是否可以刪除此叢集。如果啟DeletionProtection用，除非已修改並DeletionProtection停用叢集，否則無法刪除叢集。DeletionProtection防止叢集遭到意外刪除。

類型：布林值

必要：否

EnableCloudwatchLogsExports. 成員.

匯出至 Amazon CloudWatch 日誌時必須啟用的記錄類型清單。

類型：字串陣列

必要：否

EngineVersion

用於新叢集的資料庫引擎版本。

類型：字串

必要：否

KmsKeyId

從資料庫快照或叢集快照還原加密叢集時使用的AWS KMS金鑰識別碼。

AWS KMS 金鑰識別碼是 AWS KMS 加密金鑰的 Amazon Resource Name (ARN)。如果您要使用擁有用於加密新叢集之AWS KMS加密金鑰的叢集來還原叢集，則可以使用AWS KMS金鑰別名而非ARN 作為AWS KMS加密金鑰。AWS 帳戶

如果您未指定 KmsKeyId 參數的值，會發生以下情況：

- 如果中的快照或叢集快照SnapshotIdentifier已加密，則還原的叢集會使用用來加密快照或叢集快照的AWS KMS金鑰加密。
- 如果中的快照或叢集快照SnapshotIdentifier未加密，則還原的資料庫叢集不會加密。

類型：字串

必要：否

Port

新叢集接受連線的連接埠號碼。

限制：必須是從1150到的值65535。

預設值：與原始叢集相同的連接埠。

類型：整數

必要：否

StorageType

和資料庫叢集相關聯的儲存類型。

如需 Amazon DocumentDB 叢集的儲存類型的相關資訊，請參閱 Amazon DocumentDB 開發人員指南中的叢集儲存組態。

儲存類型的有效值-standard | iopt1

預設值為 standard

類型：字串

必要：否

標籤. N

要指派給還原叢集的標籤。

類型：[Tag](#) 物件陣列

必要：否

VpcSecurityGroupIds。 VpcSecurityGroupId.N

新叢集將屬於的虛擬私人雲端 (VPC) 安全性群組清單。

類型：字串陣列

必要：否

回應元素

服務會傳回下列元素。

DBCluster

叢集的詳細資訊。

類型：[DBCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterAlreadyExistsFault

您已經擁有具有指定識別碼的叢集。

HTTP 狀態碼：400

DBClusterQuotaExceededFault

無法建立叢集，因為您已達到叢集允許的最大配額。

HTTP 狀態碼：403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`不會參考現有的叢集快照。

HTTP 狀態碼：404

DBSnapshotNotFound

`DBSnapshotIdentifier`不參考現有的快照。

HTTP 狀態碼：404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`不參照現有的子網路群組。

HTTP 狀態碼：404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`不參照現有的子網路群組。

HTTP 狀態碼：404

InsufficientDBClusterCapacityFault

叢集沒有足夠的容量進行目前的作業。

HTTP 狀態碼：403

InsufficientStorageClusterCapacity

沒有足夠的儲存空間可用於目前動作。您可以透過將子網路群組更新為使用具有更多儲存空間的不同可用區域來解決此錯誤。

HTTP 狀態碼：400

InvalidDBClusterSnapshotStateFault

提供的值不是有效的叢集快照狀態。

HTTP 狀態碼：400

InvalidDBSnapshotState

快照的狀態不允許刪除。

HTTP 狀態碼：400

InvalidRestoreFault

您無法從虛擬私有雲 (VPC) 備份還原至非 VPC 資料庫執行個體。

HTTP 狀態碼：400

InvalidSubnet

要求的子網路無效，或要求多個子網路並非全部位於一般虛擬私人雲端 (VPC) 中。

HTTP 狀態碼：400

InvalidVPCNetworkStateFault

子網路群組在建立之後，並不會涵蓋所有可用區域，因為已進行變更。

HTTP 狀態碼：400

KMSKeyNotAccessibleFault

存取AWS KMS金鑰時發生錯誤。

HTTP 狀態碼：400

StorageQuotaExceeded

該請求將導致您超過所有實例允許的可用存儲量。

HTTP 狀態碼：400

StorageQuotaExceeded

該請求將導致您超過所有實例允許的可用存儲量。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

RestoreDBClusterToPointInTime

服務：Amazon DocumentDB (with MongoDB compatibility)

將叢集還原至任意時間點。使用者可以還原到 LatestRestorableTime 之前的任意時間點，最多 BackupRetentionPeriod 天。目標叢集是從具有與原始叢集相同組態的來源叢集建立的，但新叢集是使用預設安全性群組建立的。

請求參數

如需所有動作的一般參數相關資訊，請參閱 [《Common Parameters》](#)。

DBClusterIdentifier

要建立之新叢集的名稱。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

類型：字串

必要：是

SourceDBClusterIdentifier

還原來源之來源叢集的識別符。

約束：

- 必須符合現有 DBCluster 的識別碼。

類型：字串

必要：是

DBSubnetGroupName

要用於新叢集的子網路群組名稱。

條件約束：如果提供，則必須與現有的名稱相符 DBSubnetGroup。

範例：mySubnetgroup

類型：字串

必要：否

DeletionProtection

指定是否可以刪除此叢集。如果啟DeletionProtection用，除非已修改並DeletionProtection停用叢集，否則無法刪除叢集。DeletionProtection防止叢集遭到意外刪除。

類型：布林值

必要：否

EnableCloudwatchLogsExports. 成員.

匯出至 Amazon CloudWatch 日誌時必須啟用的記錄類型清單。

類型：字串陣列

必要：否

KmsKeyId

從加密叢集還原加密叢集時使用的AWS KMS金鑰識別碼。

AWS KMS 金鑰識別碼是 AWS KMS 加密金鑰的 Amazon Resource Name (ARN)。如果您要使用擁有用於加密新叢集之AWS KMS加密金鑰的叢集來還原叢集，則可以使用AWS KMS金鑰別名而非ARN 作為AWS KMS加密金鑰。AWS 帳戶

您可以還原到新叢集，並使用與用來加密來源叢集的金AWS KMS鑰不同的AWS KMS金鑰來加密新叢集。新的資料庫叢集會使用KmsKeyId參數所識別的AWS KMS金鑰加密。

如果您未指定 KmsKeyId 參數的值，會發生以下情況：

- 如果叢集已加密，則還原的叢集會使用用來加密來源叢集的AWS KMS金鑰加密。
- 如果叢集未加密，則不會加密還原的叢集。

如果DBClusterIdentifier指的是未加密的叢集，則會拒絕還原要求。

類型：字串

必要：否

Port

新叢集接受連線的連接埠號碼。

限制：必須是從1150到的值65535。

預設值：引擎的預設連接埠。

類型：整數

必要：否

RestoreToTime

將叢集還原至此日期和時間。

有效值：國際標準時間 (UTC) 格式的時間。

約束：

- 必須在執行個體最近一次可還原時間之前。
- 如果未提供 `UseLatestRestorableTime` 參數，則必須指定。
- 如果 `UseLatestRestorableTime` 參數為 `true`，則無法指定。
- 如果 `RestoreType` 參數為 `copy-on-write`，則無法指定。

範例：2015-03-07T23:45:00Z

類型：Timestamp

必要：否

RestoreType

要執行的還原類型。您可以指定下列其中一個值：

- `full-copy` - 新資料庫叢集將還原為來源資料庫叢集的完整複本。
- `copy-on-write` - 新資料庫叢集將還原為來源資料庫叢集的複製品。

限制：如果來源資料庫叢集的引擎版本早於 1.11，您無法指定 `copy-on-write`。

如果您不指定 `RestoreType` 值，則新資料庫叢集會還原為來源資料庫叢集的完整複本。

類型：字串

必要：否

StorageType

和資料庫叢集相關聯的儲存類型。

如需 Amazon DocumentDB 叢集的儲存類型的相關資訊，請參閱 Amazon DocumentDB 開發人員指南中的叢集儲存組態。

儲存類型的有效值-standard | iopt1

預設值為 standard

類型：字串

必要：否

標籤. N

要指派給還原叢集的標籤。

類型：[Tag](#) 物件陣列

必要：否

UseLatestRestorableTime

值設定為 true 可將叢集還原到最近可還原的備份時間，否則為 false。

預設：false

限制條件：如果提供了 RestoreToTime 參數，則無法指定。

類型：布林值

必要：否

VpcSecurityGroupIds。 VpcSecurityGroupId.N

新叢集所屬的 VPC 安全群組清單。

類型：字串陣列

必要：否

回應元素

服務會傳回下列元素。

DBCluster

叢集的詳細資訊。

類型：[DBCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterAlreadyExistsFault

您已經擁有具有指定識別碼的叢集。

HTTP 狀態碼：400

DBClusterNotFoundFault

`DBClusterIdentifier`不參考現有叢集。

HTTP 狀態碼：404

DBClusterQuotaExceededFault

無法建立叢集，因為您已達到叢集允許的最大配額。

HTTP 狀態碼：403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`不參考現有的叢集快照。

HTTP 狀態碼：404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`不參照現有的子網路群組。

HTTP 狀態碼：404

InsufficientDBClusterCapacityFault

叢集沒有足夠的容量進行目前的作業。

HTTP 狀態碼：403

InsufficientStorageClusterCapacity

沒有足夠的儲存空間可用於目前動作。您可以透過將子網路群組更新為使用具有更多儲存空間的不同可用區域來解決此錯誤。

HTTP 狀態碼：400

InvalidDBClusterSnapshotStateFault

提供的值不是有效的叢集快照狀態。

HTTP 狀態碼：400

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidDBSnapshotState

快照的狀態不允許刪除。

HTTP 狀態碼：400

InvalidRestoreFault

您無法從虛擬私有雲 (VPC) 備份還原至非 VPC 資料庫執行個體。

HTTP 狀態碼：400

InvalidSubnet

要求的子網路無效，或要求多個子網路並非全部位於一般虛擬私人雲端 (VPC) 中。

HTTP 狀態碼：400

InvalidVPCNetworkStateFault

子網路群組在建立之後，並不會涵蓋所有可用區域，因為已進行變更。

HTTP 狀態碼：400

KMSKeyNotAccessibleFault

存取AWS KMS金鑰時發生錯誤。

HTTP 狀態碼：400

StorageQuotaExceeded

該請求將導致您超過所有實例允許的可用存儲量。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

StartDBCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

重新啟動由指定的已停止叢集DBClusterIdentifier。如需詳細資訊，請參閱[停止和啟動 Amazon DocumentDB 叢集](#)。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBClusterIdentifier

要重新啟動的叢集識別碼。範例：docdb-2019-05-28-15-24-52

類型：字串

必要：是

回應元素

服務會傳回下列元素。

DBCluster

叢集的詳細資訊。

類型：[DBCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

DBClusterIdentifier不參照現有的叢集。

HTTP 狀態碼：404

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

StopDBCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

停止由指定的執行叢集DBClusterIdentifier。叢集必須處於可用狀態。如需詳細資訊，請參閱[停止和啟動 Amazon DocumentDB 叢集](#)。

請求參數

如需所有動作的一般參數相關資訊，請參閱《[Common Parameters](#)》。

DBClusterIdentifier

要停止的叢集識別碼。範例：docdb-2019-05-28-15-24-52

類型：字串

必要：是

回應元素

服務會傳回下列元素。

DBCluster

叢集的詳細資訊。

類型：[DBCluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

DBClusterNotFoundFault

DBClusterIdentifier不參考現有叢集。

HTTP 狀態碼：404

InvalidDBClusterStateFault

叢集不是有效的狀態。

HTTP 狀態碼：400

InvalidDBInstanceState

指定的執行個體不處於可用狀態。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

Amazon DocumentDB Elastic Clusters

Amazon DocumentDB 彈性叢集支援下列動作：

- [CopyClusterSnapshot](#)
- [CreateCluster](#)
- [CreateClusterSnapshot](#)
- [DeleteCluster](#)
- [DeleteClusterSnapshot](#)
- [GetCluster](#)
- [GetClusterSnapshot](#)
- [ListClusters](#)
- [ListClusterSnapshots](#)
- [ListTagsForResource](#)

- [RestoreClusterFromSnapshot](#)
- [StartCluster](#)
- [StopCluster](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateCluster](#)

CopyClusterSnapshot

服務：Amazon DocumentDB Elastic Clusters

複製彈性叢集的快照。

請求語法

```
POST /cluster-snapshot/snapshotArn/copy HTTP/1.1
Content-type: application/json
```

```
{
  "copyTags": boolean,
  "kmsKeyId": "string",
  "tags": {
    "string" : "string"
  },
  "targetSnapshotName": "string"
}
```

URI 請求參數

請求會使用下列 URI 參數。

snapshotArn

彈性叢集快照的 Amazon 資源名稱 (ARN) 識別碼。

必要：是

請求主體

請求接受採用 JSON 格式的下列資料。

targetSnapshotName

要從來源叢集快照建立之新彈性叢集快照的識別碼。此參數不區分大小寫。

約束：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：`elastic-cluster-snapshot-5`

類型：字串

長度限制：長度下限為 1。長度上限為 63。

必要：是

copyTags

設定為 `true` 可將來源叢集快照中的所有標籤複製到目標彈性叢集快照。預設值為 `false`。

類型：布林值

必要：否

kmsKeyId

加密彈性叢集快照的 AWS KMS 金鑰識別碼。AWS KMS 金鑰識別碼是 Amazon 資源名稱 (ARN)、AWS KMS 金鑰識別碼或 AWS KMS 加密金鑰的 AWS KMS 金鑰別名。

如果您從 AWS 帳戶複製加密的彈性叢集快照，您可以指定一個值，以 `KmsKeyId` 使用新的 AWS KMS 加密金鑰加密副本。如果未指定的值 `KmsKeyId`，則會使用與來源彈性叢集快照相同的 AWS KMS 金鑰來加密彈性叢集快照的副本。

若要將加密的彈性叢集快照複製 `KmsKeyId` 到其他 AWS 區域，請設定為您要用來加密目的地區域中彈性叢集快照副本的 AWS KMS 金鑰 ID。AWS KMS 加密金鑰專屬於建立金鑰的 AWS 地區，您無法使用其 AWS 他區域中某個 AWS 區域的加密金鑰。

如果您複製未加密的彈性叢集快照並指定 `KmsKeyId` 參數的值，則會傳回錯誤。

類型：字串

必要：否

tags

要指派給彈性叢集快照的標籤。

類型：字串到字串映射

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式：`^(?!aws:)[a-zA-Z+-. _:/]+$`

值長度限制：最小長度為 0。長度上限為 256。

必要：否

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "snapshot": {
    "adminUserName": "string",
    "clusterArn": "string",
    "clusterCreationTime": "string",
    "kmsKeyId": "string",
    "snapshotArn": "string",
    "snapshotCreationTime": "string",
    "snapshotName": "string",
    "snapshotType": "string",
    "status": "string",
    "subnetIds": [ "string" ],
    "vpcSecurityGroupIds": [ "string" ]
  }
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

[snapshot](#)

傳回特定彈性叢集快照的相關資訊。

類型：[ClusterSnapshot](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

ConflictException

發生存取衝突。

HTTP 狀態碼：409

InternalServerErrorException

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ServiceQuotaExceededException

超出動作的服務配額。

狀態碼：

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

CreateCluster

服務：Amazon DocumentDB Elastic Clusters

建立新的 Amazon DocumentDB 彈性叢集，並傳回其叢集結構。

請求語法

```
POST /cluster HTTP/1.1
Content-type: application/json

{
  "adminUserName": "string",
  "adminUserPassword": "string",
  "authType": "string",
  "backupRetentionPeriod": number,
  "clientToken": "string",
  "clusterName": "string",
  "kmsKeyId": "string",
  "preferredBackupWindow": "string",
  "preferredMaintenanceWindow": "string",
  "shardCapacity": number,
  "shardCount": number,
  "shardInstanceCount": number,
  "subnetIds": [ "string" ],
  "tags": {
    "string" : "string"
  },
  "vpcSecurityGroupIds": [ "string" ]
}
```

URI 請求參數

請求不會使用任何 URI 參數。

請求主體

請求接受採用 JSON 格式的下列資料。

adminUserName

Amazon DocumentDB 彈性叢集管理員的名稱。

限制條件：

- 必須為 1 到 63 個字母或數字。
- 第一個字元必須是字母。
- 不能是保留字。

類型：字串

必要：是

adminUserPassword

Amazon DocumentDB 彈性叢集管理員的密碼。密碼可以包含任何可列印的 ASCII 字元。

限制條件：

- 必須包含 8 至 100 個字元。
- 不能包含正斜線 (/)、雙引號 (") 或「at」符號 (@)。

類型：字串

必要：是

authType

用來決定從何處擷取用於存取彈性叢集之密碼的驗證類型。有效類型為PLAIN_TEXT或SECRET_ARN。

類型：字串

有效值:PLAIN_TEXT | SECRET_ARN

必要：是

clusterName

新彈性叢集的名稱。此參數是以小寫字母字串的形式儲存。

限制條件：

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：my-cluster

類型：字串

必要：是

shardCapacity

指派給每個彈性叢集碎片的 vCPUs 數目。最大值為 64 個。允許的值為 2、4、8、16、32、64。

類型：整數

必要：是

shardCount

指派給彈性叢集的碎片數目。最大值為 32。

類型：整數

必要：是

backupRetentionPeriod

自動快照的保留天數。

類型：整數

必要：否

clientToken

彈性叢集的用戶端權杖。

類型：字串

必要：否

kmsKeyId

用來加密新彈性叢集的 KMS 金鑰識別碼。

KMS 金鑰識別符是 KMS 加密金鑰的 Amazon Resource Name (ARN)。如果您使用擁有此 KMS 加密金鑰的相同 Amazon 帳戶建立叢集，則可以使用 KMS 金鑰別名而非 ARN 做為 KMS 加密金鑰。

如果未指定加密金鑰，Amazon DocumentDB 會使用 KMS 為您的帳戶建立的預設加密金鑰。您的帳戶對每個 Amazon 區域都有不同的預設加密金鑰。

類型：字串

必要：否

[preferredBackupWindow](#)

如果啟用了自動備份，則會建立自動備份的每日時間範圍 (由決定) `backupRetentionPeriod`。

類型：字串

必要：否

[preferredMaintenanceWindow](#)

每週可能進行系統維護的時段，以國際標準時間 (UTC) 表示。

格式：`ddd:hh24:mi-ddd:hh24:mi`

預設值：從每個 8 小時的時間段中隨機選取 30 分鐘的視窗 AWS 區域，發生在一週中的隨機日期。

有效日期：星期一，星期二，星期三，星期四，星期五，星期六，星期日

限制條件：必須至少是 30 分鐘的時段。

類型：字串

必要：否

[shardInstanceCount](#)

套用至彈性叢集中所有碎片的複本執行個體數目。`shardInstanceCount` 值 1 表示有一個寫入器執行個體，而任何其他執行個體都是可用於讀取和改善可用性的複本。

類型：整數

必要：否

[subnetIds](#)

新彈性叢集的 Amazon EC2 子網路 ID。

類型：字串陣列

必要：否

[tags](#)

要指派給新彈性叢集的標籤。

類型：字串到字串映射

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式：`^(?!aws:)[a-zA-Z+-. _:/]+`

值長度限制：最小長度為 0。長度上限為 256。

必要：否

[vpcSecurityGroupIds](#)

要與新彈性叢集關聯的 EC2 VPC 安全群組清單。

類型：字串陣列

必要：否

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "cluster": {
    "adminUserName": "string",
    "authType": "string",
    "backupRetentionPeriod": number,
    "clusterArn": "string",
    "clusterEndpoint": "string",
    "clusterName": "string",
    "createTime": "string",
    "kmsKeyId": "string",
    "preferredBackupWindow": "string",
    "preferredMaintenanceWindow": "string",
    "shardCapacity": number,
    "shardCount": number,
    "shardInstanceCount": number,
    "shards": [
      {
        "createTime": "string",
        "shardId": "string",
        "status": "string"
      }
    ]
  }
}
```

```
    ],  
    "status": "string",  
    "subnetIds": [ "string" ],  
    "vpcSecurityGroupIds": [ "string" ]  
  }  
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

cluster

已建立的新彈性叢集。

類型：[Cluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

ConflictException

發生存取衝突。

HTTP 狀態碼：409

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ServiceQuotaExceededException

超出動作的服務配額。

狀態碼：

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

CreateClusterSnapshot

服務：Amazon DocumentDB Elastic Clusters

建立彈性叢集的快照。

請求語法

```
POST /cluster-snapshot HTTP/1.1
Content-type: application/json
```

```
{
  "clusterArn": "string",
  "snapshotName": "string",
  "tags": {
    "string" : "string"
  }
}
```

URI 請求參數

請求不會使用任何 URI 參數。

請求主體

請求接受採用 JSON 格式的下列資料。

clusterArn

您要建立快照之彈性叢集的 ARN 識別碼。

類型：字串

必要：是

snapshotName

新彈性叢集快照的名稱。

類型：字串

長度限制：長度下限為 1。長度上限為 63。

必要：是

tags

要指派給新彈性叢集快照的標籤。

類型：字串到字串映射

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式：`^(?!aws:)[a-zA-Z+-._:/$]+`

值長度限制：最小長度為 0。長度上限為 256。

必要：否

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "snapshot": {
    "adminUserName": "string",
    "clusterArn": "string",
    "clusterCreationTime": "string",
    "kmsKeyId": "string",
    "snapshotArn": "string",
    "snapshotCreationTime": "string",
    "snapshotName": "string",
    "snapshotType": "string",
    "status": "string",
    "subnetIds": [ "string" ],
    "vpcSecurityGroupIds": [ "string" ]
  }
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

snapshot

傳回有關新彈性叢集快照的資訊。

類型：[ClusterSnapshot](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

ConflictException

發生存取衝突。

HTTP 狀態碼：409

InternalServerErrorException

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ServiceQuotaExceededException

超出動作的服務配額。

狀態碼：

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

DeleteCluster

服務：Amazon DocumentDB Elastic Clusters

刪除彈性叢集。

請求語法

```
DELETE /cluster/clusterArn HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

[clusterArn](#)

要刪除之彈性叢集的 ARN 識別碼。

必要：是

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "cluster": {
    "adminUserName": "string",
    "authType": "string",
    "backupRetentionPeriod": number,
    "clusterArn": "string",
    "clusterEndpoint": "string",
    "clusterName": "string",
    "createTime": "string",
    "kmsKeyId": "string",
    "preferredBackupWindow": "string",
    "preferredMaintenanceWindow": "string",
    "shardCapacity": number,
    "shardCount": number,
```

```
"shardInstanceCount": number,
"shards": [
  {
    "createTime": "string",
    "shardId": "string",
    "status": "string"
  }
],
"status": "string",
"subnetIds": [ "string" ],
"vpcSecurityGroupIds": [ "string" ]
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

cluster

傳回新刪除之彈性叢集的相關資訊。

類型：[Cluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

ConflictException

發生存取衝突。

HTTP 狀態碼：409

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的開發](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

DeleteClusterSnapshot

服務：Amazon DocumentDB Elastic Clusters

刪除彈性叢集快照。

請求語法

```
DELETE /cluster-snapshot/snapshotArn HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

[snapshotArn](#)

要刪除之彈性叢集快照的 ARN 識別碼。

必要：是

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "snapshot": {
    "adminUserName": "string",
    "clusterArn": "string",
    "clusterCreationTime": "string",
    "kmsKeyId": "string",
    "snapshotArn": "string",
    "snapshotCreationTime": "string",
    "snapshotName": "string",
    "snapshotType": "string",
    "status": "string",
    "subnetIds": [ "string" ],
    "vpcSecurityGroupIds": [ "string" ]
  }
}
```

```
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

[snapshot](#)

傳回新刪除之彈性叢集快照的相關資訊。

類型：[ClusterSnapshot](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

ConflictException

存取衝突。

HTTP 狀態碼：409

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的開發](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

GetCluster

服務：Amazon DocumentDB Elastic Clusters

傳回特定彈性叢集的相關資訊。

請求語法

```
GET /cluster/clusterArn HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

clusterArn

彈性叢集的 ARN 識別碼。

必要：是

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "cluster": {
    "adminUserName": "string",
    "authType": "string",
    "backupRetentionPeriod": number,
    "clusterArn": "string",
    "clusterEndpoint": "string",
    "clusterName": "string",
    "createTime": "string",
    "kmsKeyId": "string",
    "preferredBackupWindow": "string",
    "preferredMaintenanceWindow": "string",
    "shardCapacity": number,
    "shardCount": number,
```

```
"shardInstanceCount": number,
"shards": [
  {
    "createTime": "string",
    "shardId": "string",
    "status": "string"
  }
],
"status": "string",
"subnetIds": [ "string" ],
"vpcSecurityGroupIds": [ "string" ]
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

cluster

傳回特定彈性叢集的相關資訊。

類型：[Cluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

GetClusterSnapshot

服務：Amazon DocumentDB Elastic Clusters

傳回特定彈性叢集快照的相關資訊

請求語法

```
GET /cluster-snapshot/snapshotArn HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

snapshotArn

彈性叢集快照集的 ARN 識別碼。

必要：是

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "snapshot": {
    "adminUserName": "string",
    "clusterArn": "string",
    "clusterCreationTime": "string",
    "kmsKeyId": "string",
    "snapshotArn": "string",
    "snapshotCreationTime": "string",
    "snapshotName": "string",
    "snapshotType": "string",
    "status": "string",
    "subnetIds": [ "string" ],
    "vpcSecurityGroupIds": [ "string" ]
  }
}
```

```
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

[snapshot](#)

傳回特定彈性叢集快照的相關資訊。

類型：[ClusterSnapshot](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

ListClusters

服務：Amazon DocumentDB Elastic Clusters

傳回已佈建的 Amazon DocumentDB 彈性叢集的相關資訊。

請求語法

```
GET /clusters?maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

[maxResults](#)

要在回應中接收的彈性叢集快照結果數目上限。

有效範圍：最小值為 1。最大值為 100。

[nextToken](#)

先前請求提供的分頁令牌。如果指定了這個參數，則回應只會包含超出此 Token 的記錄，直到由指定的值為止max-results。

如果回應中沒有其他資料，將不nextToken會傳回。

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "clusters": [
    {
      "clusterArn": "string",
      "clusterName": "string",
      "status": "string"
    }
  ]
}
```

```
],  
  "nextToken": "string"  
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

[clusters](#)

Amazon DocumentDB 彈性集群的列表。

類型：[ClusterInList](#) 物件陣列

[nextToken](#)

先前請求提供的分頁令牌。如果指定了這個參數，則回應只會包含超出此 Token 的記錄，直到由指定的值為止max-results。

如果回應中沒有其他資料，將不nextToken會傳回。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的開發](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

ListClusterSnapshots

服務：Amazon DocumentDB Elastic Clusters

傳回指定彈性叢集之快照的相關資訊。

請求語法

```
GET /cluster-snapshots?  
clusterArn=clusterArn&maxResults=maxResults&nextToken=nextToken&snapshotType=snapshotType  
HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

[clusterArn](#)

彈性叢集的 ARN 識別碼。

[maxResults](#)

要在回應中接收的彈性叢集快照結果數目上限。

有效範圍：最小值 20。最大值為 100。

[nextToken](#)

先前請求提供的分頁令牌。如果指定了這個參數，則回應只會包含超出此 Token 的記錄，直到由指定的值為止max-results。

如果回應中沒有其他資料，將不nextToken會傳回。

[snapshotType](#)

要傳回的叢集快照類型。您可以指定下列其中一個值：

- automated-傳回 Amazon DocumentDB 為您 AWS 的帳戶自動建立的所有叢集快照。
- manual-傳回您為 AWS 帳戶手動建立的所有叢集快照。

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "nextToken": "string",
  "snapshots": [
    {
      "clusterArn": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "status": "string"
    }
  ]
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

[nextToken](#)

先前請求提供的分頁令牌。如果指定了這個參數，則回應只會包含超出此 Token 的記錄，直到由指定的值為止max-results。

如果回應中沒有其他資料，將不nextToken會傳回。

類型：字串

[snapshots](#)

指定彈性叢集的快照清單。

類型：[ClusterSnapshotInList](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

InternalServerErrorException

發生內部伺服器錯誤。

HTTP 狀態碼：500

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

ListTagsForResource

服務：Amazon DocumentDB Elastic Clusters

列出彈性叢集資源上的所有標籤

請求語法

```
GET /tags/resourceArn HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

[resourceArn](#)

彈性叢集資源的 ARN 識別碼。

長度限制：長度下限為 1。最大長度為 1011。

必要：是

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "tags": {
    "string" : "string"
  }
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

[tags](#)

指定彈性叢集資源的標籤清單。

類型：字串到字串映射

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式：`^(?!aws:)[a-zA-Z+-._:/$]`

值長度限制：最小長度為 0。長度上限為 256。

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

RestoreClusterFromSnapshot

服務：Amazon DocumentDB Elastic Clusters

從快照還原彈性叢集。

請求語法

```
POST /cluster-snapshot/snapshotArn/restore HTTP/1.1
Content-type: application/json
```

```
{
  "clusterName": "string",
  "kmsKeyId": "string",
  "shardCapacity": number,
  "shardInstanceCount": number,
  "subnetIds": [ "string" ],
  "tags": {
    "string" : "string"
  },
  "vpcSecurityGroupIds": [ "string" ]
}
```

URI 請求參數

請求會使用下列 URI 參數。

snapshotArn

彈性叢集快照集的 ARN 識別碼。

必要：是

請求主體

請求接受採用 JSON 格式的下列資料。

clusterName

彈性叢集的名稱。

類型：字串

必要：是

[kmsKeyId](#)

用來加密新 Amazon DocumentDB 彈性叢集叢集的 KMS 金鑰識別碼。

KMS 金鑰識別符是 KMS 加密金鑰的 Amazon Resource Name (ARN)。如果您使用擁有此 KMS 加密金鑰的相同 Amazon 帳戶建立叢集，則可以使用 KMS 金鑰別名而非 ARN 做為 KMS 加密金鑰。

如果未在此處指定加密金鑰，Amazon DocumentDB 會使用 KMS 為您的帳戶建立的預設加密金鑰。您的帳戶對每個 Amazon 區域都有不同的預設加密金鑰。

類型：字串

必要：否

[shardCapacity](#)

新還原的彈性叢集中每個碎片的容量。

類型：整數

必要：否

[shardInstanceCount](#)

套用至彈性叢集中所有碎片的複本執行個體數目。shardInstanceCount 值 1 表示有一個寫入器執行個體，而任何其他執行個體都是可用於讀取和改善可用性的複本。

類型：整數

必要：否

[subnetIds](#)

彈性叢集的 Amazon EC2 子網路 ID。

類型：字串陣列

必要：否

[tags](#)

要分配給恢復的彈性集群的標籤名稱的列表，以鍵值對數組的形式，其中鍵是標籤名稱，值是鍵值。

類型：字串到字串映射

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式：`^(?!aws:)[a-zA-Z+--=._:/]+$`

值長度限制：最小長度為 0。長度上限為 256。

必要：否

[vpcSecurityGroupIds](#)

要與彈性叢集關聯的 EC2 VPC 安全群組清單。

類型：字串陣列

必要：否

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "cluster": {
    "adminUserName": "string",
    "authType": "string",
    "backupRetentionPeriod": number,
    "clusterArn": "string",
    "clusterEndpoint": "string",
    "clusterName": "string",
    "createTime": "string",
    "kmsKeyId": "string",
    "preferredBackupWindow": "string",
    "preferredMaintenanceWindow": "string",
    "shardCapacity": number,
    "shardCount": number,
    "shardInstanceCount": number,
    "shards": [
      {
        "createTime": "string",
        "shardId": "string",
        "status": "string"
      }
    ],
    "status": "string",
```

```
    "subnetIds": [ "string" ],
    "vpcSecurityGroupIds": [ "string" ]
  }
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

[cluster](#)

傳回還原之彈性叢集的相關資訊。

類型：[Cluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

ConflictException

存取衝突。

HTTP 狀態碼：409

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ServiceQuotaExceededException

超出動作的服務配額。

狀態碼：

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

StartCluster

服務：Amazon DocumentDB Elastic Clusters

重新啟動由指定的已停止彈性叢集clusterARN。

請求語法

```
POST /cluster/clusterArn/start HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

clusterArn

彈性叢集的 ARN 識別碼。

必要：是

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "cluster": {
    "adminUserName": "string",
    "authType": "string",
    "backupRetentionPeriod": number,
    "clusterArn": "string",
    "clusterEndpoint": "string",
    "clusterName": "string",
    "createTime": "string",
    "kmsKeyId": "string",
    "preferredBackupWindow": "string",
    "preferredMaintenanceWindow": "string",
    "shardCapacity": number,
    "shardCount": number,
```

```
"shardInstanceCount": number,
"shards": [
  {
    "createTime": "string",
    "shardId": "string",
    "status": "string"
  }
],
"status": "string",
"subnetIds": [ "string" ],
"vpcSecurityGroupIds": [ "string" ]
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

cluster

傳回特定彈性叢集的相關資訊。

類型：[Cluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

StopCluster

服務：Amazon DocumentDB Elastic Clusters

停止由指定的執行中彈性叢集`clusterArn`。彈性叢集必須處於可用狀態。

請求語法

```
POST /cluster/clusterArn/stop HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

clusterArn

彈性叢集的 ARN 識別碼。

必要：是

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "cluster": {
    "adminUserName": "string",
    "authType": "string",
    "backupRetentionPeriod": number,
    "clusterArn": "string",
    "clusterEndpoint": "string",
    "clusterName": "string",
    "createTime": "string",
    "kmsKeyId": "string",
    "preferredBackupWindow": "string",
    "preferredMaintenanceWindow": "string",
    "shardCapacity": number,
    "shardCount": number,
```



```
"shardInstanceCount": number,
"shards": [
  {
    "createTime": "string",
    "shardId": "string",
    "status": "string"
  }
],
"status": "string",
"subnetIds": [ "string" ],
"vpcSecurityGroupIds": [ "string" ]
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

cluster

傳回特定彈性叢集的相關資訊。

類型：[Cluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

TagResource

服務：Amazon DocumentDB Elastic Clusters

將中繼資料標籤新增至彈性叢集資源

請求語法

```
POST /tags/resourceArn HTTP/1.1
Content-type: application/json
```

```
{
  "tags": {
    "string" : "string"
  }
}
```

URI 請求參數

請求會使用下列 URI 參數。

resourceArn

彈性叢集資源的 ARN 識別碼。

長度限制：長度下限為 1。最大長度為 1011。

必要：是

請求主體

請求接受採用 JSON 格式的下列資料。

tags

指派給彈性叢集資源的標籤。

類型：字串到字串映射

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式：`^(?!aws:)[a-zA-Z+-._:/$]+`

值長度限制：最小長度為 0。長度上限為 256。

必要：是

回應語法

```
HTTP/1.1 200
```

回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

InternalServerError

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

UntagResource

服務：Amazon DocumentDB Elastic Clusters

從彈性叢集資源移除中繼資料標籤

請求語法

```
DELETE /tags/resourceArn?tagKeys=tagKeys HTTP/1.1
```

URI 請求參數

請求會使用下列 URI 參數。

resourceArn

彈性叢集資源的 ARN 識別碼。

長度限制：長度下限為 1。最大長度為 1011。

必要：是

tagKeys

要從彈性叢集資源移除的標籤鍵。

陣列成員：項目數下限為 0。項目數上限為 50。

長度限制：長度下限為 1。長度上限為 128。

模式：`^(?!aws:)[a-zA-Z+--._:/]+$`

必要：是

請求主體

請求沒有請求主體。

回應語法

```
HTTP/1.1 200
```

回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

InternalServerErrorException

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

UpdateCluster

服務：Amazon DocumentDB Elastic Clusters

修改彈性叢集。這包括更新管理員用戶名/密碼，升級 API 版本，以及設置備份窗口和維護窗口

請求語法

```
PUT /cluster/clusterArn HTTP/1.1
Content-type: application/json

{
  "adminUserPassword": "string",
  "authType": "string",
  "backupRetentionPeriod": number,
  "clientToken": "string",
  "preferredBackupWindow": "string",
  "preferredMaintenanceWindow": "string",
  "shardCapacity": number,
  "shardCount": number,
  "shardInstanceCount": number,
  "subnetIds": [ "string" ],
  "vpcSecurityGroupIds": [ "string" ]
}
```

URI 請求參數

請求會使用下列 URI 參數。

clusterArn

彈性叢集的 ARN 識別碼。

必要：是

請求主體

請求接受採用 JSON 格式的下列資料。

adminUserPassword

與彈性叢集管理員相關聯的密碼。此密碼可包含任何可印出的 ASCII 字元，但斜線 (/)、雙引號 (") 或 "at" 符號 (@) 則除外。

限制條件：必須包含 8 到 100 個字元。

類型：字串

必要：否

authType

用來決定從何處擷取用於存取彈性叢集之密碼的驗證類型。有效類型為PLAIN_TEXT或SECRET_ARN。

類型：字串

有效值:PLAIN_TEXT | SECRET_ARN

必要：否

backupRetentionPeriod

自動快照的保留天數。

類型：整數

必要：否

clientToken

彈性叢集的用戶端權杖。

類型：字串

必要：否

preferredBackupWindow

如果啟用了自動備份，則會建立自動備份的每日時間範圍 (由決定) backupRetentionPeriod。

類型：字串

必要：否

preferredMaintenanceWindow

每週可能進行系統維護的時段，以國際標準時間 (UTC) 表示。

格式：ddd:hh24:mi-ddd:hh24:mi

預設值：從每個 8 小時的時間段中隨機選取 30 分鐘的視窗 AWS 區域，發生在一週中的隨機日期。

有效日期：星期一，星期二，星期三，星期四，星期五，星期六，星期日

限制條件：必須至少是 30 分鐘的時段。

類型：字串

必要：否

shardCapacity

指派給每個彈性叢集碎片的 vCPUs 數目。最大值為 64 個。允許的值為 2、4、8、16、32、64。

類型：整數

必要：否

shardCount

指派給彈性叢集的碎片數目。最大值為 32。

類型：整數

必要：否

shardInstanceCount

套用至彈性叢集中所有碎片的複本執行個體數目。shardInstanceCount 值 1 表示有一個寫入器執行個體，而任何其他執行個體都是可用於讀取和改善可用性的複本。

類型：整數

必要：否

subnetIds

彈性叢集的 Amazon EC2 子網路 ID。

類型：字串陣列

必要：否

vpcSecurityGroupIds

要與彈性叢集關聯的 EC2 VPC 安全群組清單。

類型：字串陣列

必要：否

回應語法

```
HTTP/1.1 200
Content-type: application/json

{
  "cluster": {
    "adminUserName": "string",
    "authType": "string",
    "backupRetentionPeriod": number,
    "clusterArn": "string",
    "clusterEndpoint": "string",
    "clusterName": "string",
    "createTime": "string",
    "kmsKeyId": "string",
    "preferredBackupWindow": "string",
    "preferredMaintenanceWindow": "string",
    "shardCapacity": number,
    "shardCount": number,
    "shardInstanceCount": number,
    "shards": [
      {
        "createTime": "string",
        "shardId": "string",
        "status": "string"
      }
    ],
    "status": "string",
    "subnetIds": [ "string" ],
    "vpcSecurityGroupIds": [ "string" ]
  }
}
```

回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

[cluster](#)

傳回有關更新彈性叢集的資訊。

類型：[Cluster](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

當沒有足夠的權限執行動作時，就會發生例外狀況。

HTTP 狀態碼：403

ConflictException

發生存取衝突。

HTTP 狀態碼：409

InternalServerErrorException

發生內部伺服器錯誤。

HTTP 狀態碼：500

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

ThrottlingException

ThrottlingException 當請求被拒絕時，由於請求限制將被拋出。

HTTP 狀態碼：429

ValidationException

定義驗證例外狀況的結構。

HTTP 狀態碼：400

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

資料類型

支援下列資料類型 Amazon DocumentDB (with MongoDB compatibility)：

- [AvailabilityZone](#)
- [Certificate](#)
- [CertificateDetails](#)
- [CloudwatchLogsExportConfiguration](#)
- [DBCluster](#)
- [DBClusterMember](#)
- [DBClusterParameterGroup](#)
- [DBClusterRole](#)
- [DBClusterSnapshot](#)
- [DBClusterSnapshotAttribute](#)
- [DBClusterSnapshotAttributesResult](#)
- [DBEngineVersion](#)
- [DBInstance](#)
- [DBInstanceStatusInfo](#)
- [DBSubnetGroup](#)

- [Endpoint](#)
- [EngineDefaults](#)
- [Event](#)
- [EventCategoriesMap](#)
- [EventSubscription](#)
- [Filter](#)
- [GlobalCluster](#)
- [GlobalClusterMember](#)
- [OrderableDBInstanceOption](#)
- [Parameter](#)
- [PendingCloudwatchLogsExports](#)
- [PendingMaintenanceAction](#)
- [PendingModifiedValues](#)
- [ResourcePendingMaintenanceActions](#)
- [Subnet](#)
- [Tag](#)
- [UpgradeTarget](#)
- [VpcSecurityGroupMembership](#)

下列資料類型受 Amazon DocumentDB 彈性叢集支援：

- [Cluster](#)
- [ClusterInList](#)
- [ClusterSnapshot](#)
- [ClusterSnapshotInList](#)
- [Shard](#)
- [ValidationExceptionField](#)

Amazon DocumentDB (with MongoDB compatibility)

支援下列資料類型Amazon DocumentDB (with MongoDB compatibility)：

- [AvailabilityZone](#)
- [Certificate](#)
- [CertificateDetails](#)
- [CloudwatchLogsExportConfiguration](#)
- [DBCluster](#)
- [DBClusterMember](#)
- [DBClusterParameterGroup](#)
- [DBClusterRole](#)
- [DBClusterSnapshot](#)
- [DBClusterSnapshotAttribute](#)
- [DBClusterSnapshotAttributesResult](#)
- [DBEngineVersion](#)
- [DBInstance](#)
- [DBInstanceStatusInfo](#)
- [DBSubnetGroup](#)
- [Endpoint](#)
- [EngineDefaults](#)
- [Event](#)
- [EventCategoriesMap](#)
- [EventSubscription](#)
- [Filter](#)
- [GlobalCluster](#)
- [GlobalClusterMember](#)
- [OrderableDBInstanceOption](#)
- [Parameter](#)
- [PendingCloudwatchLogsExports](#)
- [PendingMaintenanceAction](#)
- [PendingModifiedValues](#)
- [ResourcePendingMaintenanceActions](#)
- [Subnet](#)

- [Tag](#)
- [UpgradeTarget](#)
- [VpcSecurityGroupMembership](#)

AvailabilityZone

服務: Amazon DocumentDB (with MongoDB compatibility)

可用區域的相關資訊。

內容

Note

在下列清單中，會先說明必要的參數。

Name

可用區域的名稱。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

Certificate

服務: Amazon DocumentDB (with MongoDB compatibility)

憑證授權機構 (CA) 憑證AWS 帳戶。

內容

Note

在下列清單中，會先說明必要的參數。

CertificateArn

憑證的 Amazon Resource Name (ARN)。

範例：`arn:aws:rds:us-east-1::cert:rds-ca-2019`

類型：字串

必要：否

CertificateIdentifier

可識別憑證的唯一金鑰。

範例：`rds-ca-2019`

類型：字串

必要：否

CertificateType

憑證的類型。

範例：`CA`

類型：字串

必要：否

Thumbprint

憑證的指紋。

類型：字串

必要：否

ValidFrom

憑證有效之後的開始日期和時間。

範例：2019-07-31T17:57:09Z

類型：Timestamp

必要：否

ValidTill

憑證不再有效之後的日期和時間。

範例：2024-07-31T17:57:09Z

類型：Timestamp

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

CertificateDetails

服務: Amazon DocumentDB (with MongoDB compatibility)

傳回資料庫執行個體伺服器憑證的詳細資訊。

如需詳細資訊，請參閱 [Amazon DocumentDB 開發人員指南中的更新您的 Amazon 文件資料庫 TLS 憑證和加密傳輸中的資料](#)。

目錄

Note

在下列清單中，會先說明必要的參數。

CAIdentifier

用於資料庫執行個體伺服器憑證之 CA 憑證的 CA 識別碼。

類型：字串

必要：否

ValidTill

資料庫執行個體伺服器憑證的到期日。

類型：Timestamp

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

CloudwatchLogsExportConfiguration

服務: Amazon DocumentDB (with MongoDB compatibility)

資料庫類型要匯出至 Amazon CloudWatch Logs 的組態設定。

EnableLogTypes和DisableLogTypes陣列會決定哪些記錄要匯出 (或不匯出) 至 CloudWatch 記錄。這些陣列中的值取決於正在使用的引擎。

內容

Note

在下列清單中，會先說明必要的參數。

DisableLogTypes.member.N

要停用的日誌類型清單。

類型：字串陣列

必要：否

EnableLogTypes.member.N

要啟用的日誌類型清單。

類型：字串陣列

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBCluster

服務：Amazon DocumentDB (with MongoDB compatibility)

叢集的詳細資訊。

目錄

Note

在下列清單中，會先說明必要的參數。

AssociatedRoles.DBClusterRole.N

提供與叢集相關聯的 AWS Identity and Access Management (IAM) 角色清單。(IAM) 與叢集關聯的角色授予叢集代表您存取其他AWS服務的權限。

類型：[DBClusterRole](#) 物件陣列

必要：否

AvailabilityZones.AvailabilityZone.N

提供可在叢集中建立執行個體的 Amazon EC2 可用區域清單。

類型：字串陣列

必要：否

BackupRetentionPeriod

指定保留自動快照的天數。

類型：整數

必要：否

CloneGroupId

識別與資料庫叢集建立關聯的複製群組。

類型：字串

必要：否

ClusterCreateTime

指定建立叢集的時間 (以世界標準時間 (UTC) 表示)。

類型：Timestamp

必要：否

DBClusterArn

群集的 Amazon 資源名稱 (ARN)。

類型：字串

必要：否

DBClusterIdentifier

包含使用者提供的叢集識別碼。此識別碼是識別叢集的唯一金鑰。

類型：字串

必要：否

DBClusterMembers.DBClusterMember.N

提供組成叢集的執行個體清單。

類型：[DBClusterMember](#) 物件陣列

必要：否

DBClusterParameterGroup

指定叢集的叢集參數群組名稱。

類型：字串

必要：否

DbClusterResourceId

叢AWS 區域集的唯一、不可變的識別碼。每當存取叢集的AWS KMS金鑰時，都會在AWS CloudTrail記錄項目中找到此識別碼。

類型：字串

必要：否

DBSubnetGroup

指定與叢集相關聯之子網路群組的資訊，包括子網路群組中的名稱、說明和子網路。

類型：字串

必要：否

DeletionProtection

指定是否可以刪除此叢集。如果啟DeletionProtection用，除非已修改並DeletionProtection停用叢集，否則無法刪除叢集。DeletionProtection防止叢集遭到意外刪除。

類型：布林值

必要：否

EarliestRestorableTime

使用還原功能可以 point-in-time還原資料庫的最早時間。

類型：Timestamp

必要：否

EnabledCloudwatchLogsExports.member.N

此叢集設定為匯出至 Amazon CloudWatch 日誌的日誌類型清單。

類型：字串陣列

必要：否

Endpoint

指定叢集主要執行個體的連線端點。

類型：字串

必要：否

Engine

提供要用於此叢集的資料庫引擎名稱。

類型：字串

必要：否

EngineVersion

指示資料庫引擎版本。

類型：字串

必要：否

HostedZoneId

指定當您建立託管區域時，Amazon Route 53 指派的 ID。

類型：字串

必要：否

KmsKeyId

如果StorageEncrypted是true，則為加密叢集的AWS KMS金鑰識別碼。

類型：字串

必要：否

LatestRestorableTime

指定可以透過還原來 point-in-time還原資料庫的最近時間。

類型：Timestamp

必要：否

MasterUsername

包含叢集的主要使用者名稱。

類型：字串

必要：否

MultiAZ

指定叢集是否有多個可用區域中的執行個體。

類型：布林值

必要：否

PercentProgress

指定操作的進度 (百分比)。

類型：字串

必要：否

Port

指定資料庫引擎接聽的連接埠。

類型：整數

必要：否

PreferredBackupWindow

指定每日的時間範圍，若有啟用自動化備份 (由 BackupRetentionPeriod 決定)，則自動化備份會在此期間建立。

類型：字串

必要：否

PreferredMaintenanceWindow

指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。

類型：字串

必要：否

ReaderEndpoint

叢集的讀取器端點。叢集的讀取器端點負載會平衡叢集中可用之 Amazon DocumentDB 複本之間的連線。當用戶端要求與讀取器端點的新連線時，Amazon DocumentDB 會在叢集中的 Amazon DocumentDB 複本之間分散連線請求。此功能有助於平衡叢集中多個 Amazon DocumentDB 複本的讀取工作負載。

如果發生容錯移轉，且您所連線的 Amazon DocumentDB 複本被提升為主執行個體，則您的連線會中斷。若要繼續將讀取工作負載傳送到叢集中的其他 Amazon DocumentDB 複本，您可以接著重新連線到讀取器端點。

類型：字串

必要：否

ReadReplicaIdentifiers.ReadReplicaIdentifier.N

包含與此叢集相關聯之次要叢集之一或多個識別碼。

類型：字串陣列

必要：否

ReplicationSourceIdentifier

如果此叢集是次要叢集，則包含來源叢集的識別碼。

類型：字串

必要：否

Status

指定此叢集的目前狀態。

類型：字串

必要：否

StorageEncrypted

指定是否要加密叢集。

類型：布林值

必要：否

StorageType

與叢集相關聯的儲存區類型

與叢集相關聯的儲存區類型

如需 Amazon DocumentDB 叢集的儲存類型的相關資訊，請參閱 Amazon DocumentDB 開發人員指南中的叢集儲存組態。

儲存類型的有效值-`standard` | `iopt1`

預設值為 standard

類型：字串

必要：否

VpcSecurityGroups.VpcSecurityGroupMembership.N

提供叢集所屬的虛擬私人雲端 (VPC) 安全性群組清單。

類型：[VpcSecurityGroupMembership](#) 物件陣列

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBClusterMember

服務: Amazon DocumentDB (with MongoDB compatibility)

包含屬於叢集一部分之執行個體的相關資訊。

內容

Note

在下列清單中，會先說明必要的參數。

DBClusterParameterGroupStatus

針對資料庫叢集設定此參數群組。

類型：字串

必要：否

DBInstanceIdentifier

指定叢集此成員的執行個體識別碼。

類型：字串

必要：否

IsClusterWriter

一個值，表true示叢集成員是叢集的主要執行個體，false否則為此值。

類型：布林值

必要：否

PromotionTier

此值指定在現有主要執行個體失敗後，Amazon DocumentDB 複本提升為主要執行個體的順序。

類型：整數

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBClusterParameterGroup

服務: Amazon DocumentDB (with MongoDB compatibility)

叢集參數群組的詳細資訊。

內容

Note

在下列清單中，會先說明必要的參數。

DBClusterParameterGroupArn

叢集參數群組的 Amazon Resource Name (ARN)。

類型：字串

必要：否

DBClusterParameterGroupName

提供叢集參數群組的名稱。

類型：字串

必要：否

DBParameterGroupFamily

提供與此叢集參數群組相容的參數群組系列名稱。

類型：字串

必要：否

Description

提供與此叢集參數群組的客戶指定描述。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBClusterRole

服務: Amazon DocumentDB (with MongoDB compatibility)

描述與叢集相關聯的AWS Identity and Access Management (IAM) 角色。

內容

Note

在下列清單中，會先說明必要的參數。

RoleArn

與資料庫叢集相關聯的 iamResource Name (ARN)。

類型：字串

必要：否

Status

說明 iamRole 和叢集之間的關聯狀態。該Status屬性返回下列其中一個值：

- ACTIVE-iamRole ARN 與叢集相關聯，可用來代表您存取其他AWS服務。
- PENDING-iamRole ARN 正在與叢集相關聯。
- INVALID-iamRole ARN 與叢集相關聯，但叢集無法假設 IAMRole 代表您存取其他AWS服務。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBClusterSnapshot

服務：Amazon DocumentDB (with MongoDB compatibility)

叢集快照的詳細資訊。

目錄

Note

在下列清單中，會先說明必要的參數。

AvailabilityZones.AvailabilityZone.N

提供可在叢集快照中還原執行個體的 Amazon EC2 可用區域清單。

類型：字串陣列

必要：否

ClusterCreateTime

指定建立叢集的時間 (以世界標準時間 (UTC) 表示)。

類型：Timestamp

必要：否

DBClusterIdentifier

指定建立此叢集快照之叢集的叢集識別碼。

類型：字串

必要：否

DBClusterSnapshotArn

叢集快照的 Amazon 資源名稱 (ARN)。

類型：字串

必要：否

DBClusterSnapshotIdentifier

指定叢集快照的識別碼。

類型：字串

必要：否

Engine

指定資料庫引擎的名稱。

類型：字串

必要：否

EngineVersion

提供此叢集快照集的資料庫引擎版本。

類型：字串

必要：否

KmsKeyId

如果StorageEncrypted是true，則為加密叢集快照的AWS KMS金鑰識別碼。

類型：字串

必要：否

MasterUsername

提供叢集快照的主要使用者名稱。

類型：字串

必要：否

PercentProgress

指定已傳輸的估計資料百分比。

類型：整數

必要：否

Port

指定快照時叢集偵聽的連接埠。

類型：整數

必要：否

SnapshotCreateTime

提供拍攝快照的時間 (以 UTC 為單位)。

類型：Timestamp

必要：否

SnapshotType

提供叢集快照的類型。

類型：字串

必要：否

SourceDBClusterSnapshotArn

如果叢集快照是從來源叢集快照複製的，則為來源叢集快照的 ARN，否則為 null 值。

類型：字串

必要：否

Status

指定此叢集快照的狀態。

類型：字串

必要：否

StorageEncrypted

指定叢集快照是否已加密。

類型：布林值

必要：否

StorageType

與叢集快照相關聯的儲存區類型

如需 Amazon DocumentDB 叢集的儲存類型的相關資訊，請參閱 Amazon DocumentDB 開發人員指南中的叢集儲存組態。

儲存類型的有效值-standard | iopt1

預設值為 standard

類型：字串

必要：否

VpcId

提供與叢集快照相關聯的虛擬私有雲 (VPC) ID。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBClusterSnapshotAttribute

服務: Amazon DocumentDB (with MongoDB compatibility)

包含手動叢集快照屬性的名稱和值。

手動叢集快照屬性是用來授權其他AWS 帳戶人還原手動叢集快照。

內容

Note

在下列清單中，會先說明必要的參數。

AttributeName

手動叢集快照屬性的名稱。

名為的屬性restore是指具有複製或還原手動叢集快照之AWS 帳戶權限的清單。

類型：字串

必要：否

AttributeValues.AttributeValue.N

手動叢集快照屬性的值。

如果AttributeName欄位設定為restore，則此元素會傳回已授權複製或還原手動叢集快照的 ID 清單。AWS 帳戶如果清單中all有的值，則手動叢集快照為公用且可供任AWS 帳戶何人複製或還原。

類型：字串陣列

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBClusterSnapshotAttributesResult

服務: Amazon DocumentDB (with MongoDB compatibility)

與叢集快照相關聯之屬性的詳細資訊。

內容

Note

在下列清單中，會先說明必要的參數。

DBClusterSnapshotAttributes.DBClusterSnapshotAttribute.N

叢集快照的屬性和值清單。

類型：[DBClusterSnapshotAttribute](#) 物件陣列

必要：否

DBClusterSnapshotIdentifier

屬性所套用之叢集快照的識別碼。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBEngineVersion

服務: Amazon DocumentDB (with MongoDB compatibility)

引擎版本的詳細資訊。

目錄

Note

在下列清單中，會先說明必要的參數。

DBEngineDescription

資料庫引擎描述。

類型：字串

必要：否

DBEngineVersionDescription

資料庫引擎版本描述。

類型：字串

必要：否

DBParameterGroupFamily

資料庫引擎的參數群組族群名稱。

類型：字串

必要：否

Engine

資料庫引擎名稱。

類型：字串

必要：否

EngineVersion

資料庫引擎版本編號。

類型：字串

必要：否

ExportableLogTypes.member.N

資料庫引擎可匯出至 Amazon CloudWatch 日誌的日誌類型。

類型：字串陣列

必要：否

SupportedCACertificateIdentifiers.member.N

支援的 CA 憑證識別碼清單。

如需詳細資訊，請參閱 [Amazon DocumentDB 開發人員指南中的更新您的 Amazon 文件資料庫 TLS 憑證和加密傳輸中的資料](#)。

類型：字串陣列

必要：否

SupportsCertificateRotationWithoutRestart

指出引擎版本是否支援旋轉伺服器憑證，而不重新啟動資料庫執行個體。

類型：布林值

必要：否

SupportsLogExportsToCloudwatchLogs

指出引擎版本是否支援將指定的記錄類型匯出ExportableLogTypes至 CloudWatch 記錄檔的值。

類型：布林值

必要：否

ValidUpgradeTarget.UpgradeTarget.N

此資料庫引擎版本可升級的引擎版本清單。

類型：[UpgradeTarget](#) 物件陣列

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBInstance

服務: Amazon DocumentDB (with MongoDB compatibility)

執行個體的詳細資訊。

目錄

Note

在下列清單中，會先說明必要的參數。

AutoMinorVersionUpgrade

不適用。此參數不適用於 Amazon DocumentDB。無論設定的值為何，Amazon DocumentDB 都不會執行次要版本升級。

類型：布林值

必要：否

AvailabilityZone

指定執行個體所在的可用區域名稱。

類型：字串

必要：否

BackupRetentionPeriod

指定保留自動快照的天數。

類型：整數

必要：否

CACertificateIdentifier

此資料庫執行個體的憑證授權機構憑證識別碼。

類型：字串

必要：否

CertificateDetails

資料庫執行個體伺服器憑證的詳細資訊。

類型：[CertificateDetails](#) 物件

必要：否

CopyTagsToSnapshot

值，指示是否將標籤從資料庫執行個體複製到資料庫執行個體快照。根據預設，不會複製標籤。

類型：布林值

必要：否

DBClusterIdentifier

如果執行處理是叢集的成員，則包含執行處理所屬的叢集名稱。

類型：字串

必要：否

DBInstanceArn

執行個體的 Amazon Resource Name (ARN)。

類型：字串

必要：否

DBInstanceClass

包含執行個體的運算和記憶體容量類別的名稱。

類型：字串

必要：否

DBInstanceIdentifier

包含使用者提供的資料庫識別碼。此識別碼是識別執行個體的唯一金鑰。

類型：字串

必要：否

DBInstanceStatus

指定此資料庫的目前狀態。

類型：字串

必要：否

DbiResourceId

執行個AWS 區域體的唯一且不可變的識別碼。每當存取執行個體的AWS KMS金鑰時，都會在AWS CloudTrail記錄項目中找到此識別碼。

類型：字串

必要：否

DBSubnetGroup

指定與執行個體相關聯之子網路群組的資訊，包括子網路群組中的名稱、說明和子網路。

類型：[DBSubnetGroup](#) 物件

必要：否

EnabledCloudwatchLogsExports.member.N

此執行個體設定為匯出至記錄檔的 CloudWatch 記錄類型清單。

類型：字串陣列

必要：否

Endpoint

指定連線端點。

類型：[Endpoint](#) 物件

必要：否

Engine

提供要用於此執行處理的資料庫引擎名稱。

類型：字串

必要：否

EngineVersion

指示資料庫引擎版本。

類型：字串

必要：否

InstanceCreateTime

提供執行個體的建立日期和時間。

類型：Timestamp

必要：否

KmsKeyId

如果StorageEncrypted是true，則為加密執行個體的AWS KMS金鑰識別碼。

類型：字串

必要：否

LatestRestorableTime

指定可以透過還原來 point-in-time還原資料庫的最近時間。

類型：Timestamp

必要：否

PendingModifiedValues

指定執行個體的變更處於擱置狀態。只有當變更處於擱置狀態時，才會包含此元素。特定的變更會由子元素識別。

類型：[PendingModifiedValues](#) 物件

必要：否

PreferredBackupWindow

指定每日的時間範圍，若有啟用自動化備份 (由 BackupRetentionPeriod 決定)，則自動化備份會在此期間建立。

類型：字串

必要：否

PreferredMaintenanceWindow

指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。

類型：字串

必要：否

PromotionTier

一個值，指定 Amazon DocumentDB 複本在現有主執行個體發生故障後提升為主執行個體的順序。

類型：整數

必要：否

PubliclyAccessible

不支援. Amazon DocumentDB 目前不支援公有端點。的值永遠PubliclyAccessible是false。

類型：布林值

必要：否

StatusInfos.DBInstanceStatusInfo.N

僅供讀取複本的狀態。如果執行個體不是僅供讀取複本，這是空白的。

類型：[DBInstanceStatusInfo](#) 物件陣列

必要：否

StorageEncrypted

指定執行個體是否已加密。

類型：布林值

必要：否

VpcSecurityGroups.VpcSecurityGroupMembership.N

提供執行個體所屬的 VPC 安全群組元素清單。

類型：[VpcSecurityGroupMembership](#) 物件陣列

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBInstanceStatusInfo

服務: Amazon DocumentDB (with MongoDB compatibility)

提供執行處理的狀態資訊清單。

內容

Note

在下列清單中，會先說明必要的參數。

Message

若執行個體發生錯誤，則為錯誤的詳細資訊。若執行個體並未處於錯誤狀態，則此值為空白。

類型：字串

必要：否

Normal

Boolean 值，表true示執行個體正常運作，或執false行個體處於錯誤狀態時。

類型：布林值

必要：否

Status

執行個體的状态。對於StatusType僅供讀取複本，值可以是replicating、錯誤stopped、或terminated。

類型：字串

必要：否

StatusType

此值目前為 "read replication。"

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

DBSubnetGroup

服務: Amazon DocumentDB (with MongoDB compatibility)

有關子網路群組的詳細資訊。

內容

Note

在下列清單中，會先說明必要的參數。

DBSubnetGroupArn

資料庫子網路群組的 Amazon Resource Name (ARN)。

類型：字串

必要：否

DBSubnetGroupDescription

提供子網路群組的描述。

類型：字串

必要：否

DBSubnetGroupName

子網路群組的名稱。

類型：字串

必要：否

SubnetGroupStatus

提供子網路群組的狀態。

類型：字串

必要：否

Subnets.Subnet.N

子網路群組的一或多個子網路的詳細資訊。

類型：[Subnet](#) 物件陣列

必要：否

VpcId

Virtual Private Cloud (VPC) 的 ID。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

Endpoint

服務: Amazon DocumentDB (with MongoDB compatibility)

存取叢集或執行個體的網路資訊。用戶端程式必須指定有效的端點才能存取這些 Amazon DocumentDB 資源。

內容

Note

在下列清單中，會先說明必要的參數。

Address

指定例行的 DNS 地址。

類型：字串

必要：否

HostedZoneId

指定當您建立託管區域時，Amazon Route 53 指派的 ID。

類型：字串

必要：否

Port

指定資料庫引擎接聽的連接埠。

類型：整數

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

EngineDefaults

服務: Amazon DocumentDB (with MongoDB compatibility)

包含成功呼叫DescribeEngineDefaultClusterParameters作業的結果。

內容

Note

在下列清單中，會先說明必要的參數。

DBParameterGroupFamily

要與此相容的叢集參數群組系列名稱。

類型：字串

必要：否

Marker

前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到MaxRecords 指定的值為止。

類型：字串

必要：否

Parameters.Parameter.N

特定叢集參數群組族群的參數。

類型：[Parameter](#) 物件陣列

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

Event

服務: Amazon DocumentDB (with MongoDB compatibility)

事件的地的相關資訊。

目錄

Note

在下列清單中，會先說明必要的參數。

Date

指定事件的日期和時間。

類型：Timestamp

必要：否

EventCategories.EventCategory.N

指定事件類別。

類型：字串陣列

必要：否

Message

提供此事件的文字。

類型：字串

必要：否

SourceArn

事件的 Amazon Resource Name (ARN)。

類型：字串

必要：否

SourceIdentifier

提供事件來源的識別符。

類型：字串

必要：否

SourceType

指定此事件的來源類型。

類型：字串

有效值: db-instance | db-parameter-group | db-security-group | db-snapshot
| db-cluster | db-cluster-snapshot

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

EventCategoriesMap

服務: Amazon DocumentDB (with MongoDB compatibility)

事件來源類型，伴隨著一或多個事件類別名稱。

內容

Note

在下列清單中，會先說明必要的參數。

EventCategories.EventCategory.N

指定來源類型的事件類別。

類型：字串陣列

必要：否

SourceType

返回的類別所屬的源類型。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

EventSubscription

服務: Amazon DocumentDB (with MongoDB compatibility)

您已訂閱的事件的詳細資訊。

內容

Note

在下列清單中，會先說明必要的參數。

CustomerAwsId

與 Amazon DocumentDB 事件通知訂閱相關聯的AWS客戶帳戶。

類型：字串

必要：否

CustSubscriptionId

Amazon DocumentDB 事件通知訂閱識別碼。

類型：字串

必要：否

Enabled

指示訂閱是否已啟用的布林值。的值表true示已啟用訂閱。

類型：布林值

必要：否

EventCategoriesList.EventCategory.N

亞馬遜事件通知訂閱的清單。

類型：字串陣列

必要：否

EventSubscriptionArn

事件訂閱的 Amazon Resource Name (ARN)。

類型：字串

必要：否

SnsTopicArn

Amazon DocumentDB 事件通知訂閱的主題 ARN。

類型：字串

必要：否

SourceIdsList.SourceId.N

Amazon 文件資料庫事件通知訂閱的來源 ID 清單。

類型：字串陣列

必要：否

SourceType

Amazon 文件資料庫事件通知訂閱的來源類型。

類型：字串

必要：否

Status

亞馬遜文件資料庫事件通知訂閱的狀態。

約束：

可以是下列其中一項：`creating`、`modifying`、`deleting`、`active`、`no-permission`、`topic-not-exist`

狀 `no-permission` 態表示 Amazon DocumentDB 不再具有張貼至 SNS 主題的權限。狀 `topic-not-exist` 態表示主題在建立訂閱之後已刪除。

類型：字串

必要：否

SubscriptionCreationTime

事 Amazon DocumentDB 通知訂閱建立的時間。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

Filter

服務: Amazon DocumentDB (with MongoDB compatibility)

一組命名的過濾器值，用於返回更具體的結果列表。您可以使用篩選器，依特定準則 (例如 ID) 來比對一組資源。

篩選條件中不支援萬用字元。

內容

Note

在下列清單中，會先說明必要的參數。

Name

篩選條件的名稱。篩選條件名稱區分大小寫。

類型：字串

必要：是

Values.Value.N

一或多個篩選條件值。篩選條件值區分大小寫。

類型：字串陣列

必要：是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

GlobalCluster

服務: Amazon DocumentDB (with MongoDB compatibility)

代表 Amazon 文件資料庫全域叢集的資料類型。

內容

Note

在下列清單中，會先說明必要的參數。

DatabaseName

新全域叢集中的預設資料庫名稱。

類型：字串

必要：否

DeletionProtection

新的全域叢集的刪除保護設定義。

類型：布林值

必要：否

Engine

全域叢集所使用的亞馬遜文件資料庫引擎。

類型：字串

必要：否

EngineVersion

指示資料庫引擎版本。

類型：字串

必要：否

GlobalClusterArn

全域叢集的 Amazon 資源 (ARN)。

類型：字串

必要：否

GlobalClusterIdentifier

包含使用者提供的全域叢集識別碼。此識別全域叢集的唯一金鑰。

類型：字串

長度限制：長度下限為 1。長度上限為 255。

模式：`[A-Za-z][0-9A-Za-z-:._]*`

必要：否

GlobalClusterMembers.GlobalClusterMember.N

全域叢集中次要叢集的叢集 ID 清單。目前僅限於一個項目。

類型：[GlobalClusterMember](#) 物件陣列

必要：否

GlobalClusterResourceid

全域資料庫叢集的AWS 區域唯一、固定識別符。此識別碼可在每次存取叢集的AWS KMS客戶主金鑰 (CMK) 時，於AWS CloudTrail日誌項目中找到。

類型：字串

必要：否

Status

指定此全域叢集的目前狀態。

類型：字串

必要：否

StorageEncrypted

全域叢集的儲存加密設定識別。

類型：布林值

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

GlobalClusterMember

服務: Amazon DocumentDB (with MongoDB compatibility)

一種資料結構，其中包含與 Amazon DocumentDB 全域叢集相關聯的任何主要和次要叢集的相關資訊。

內容

Note

在下列清單中，會先說明必要的參數。

DBClusterArn

每個 Amazon Resource Name (ARN) Amazon DocumentDB ARN)。

類型：字串

必要：否

IsWriter

指定 Amazon DocumentDB 叢集是否為與其相關聯之 Amazon DocumentDB 全域叢集的主要叢集 (亦即具有讀寫功能)。

類型：布林值

必要：否

Readers.member.N

與 Aurora ource Name (ARN) 相關聯的每個唯讀次要叢集 Amazon Resource Name (ARN)。

類型：字串陣列

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)

- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

OrderableDBInstanceOption

服務: Amazon DocumentDB (with MongoDB compatibility)

可用於執行個體的選項。

內容

Note

在下列清單中，會先說明必要的參數。

AvailabilityZones.AvailabilityZone.N

執行個體的可用區域清單。

類型：[AvailabilityZone](#) 物件陣列

必要：否

DBInstanceClass

執行個體的執行個體類別。

類型：字串

必要：否

Engine

執行個體的引擎類型。

類型：字串

必要：否

EngineVersion

執行個體的引擎版本。

類型：字串

必要：否

LicenseModel

執行個體的授權模式。

類型：字串

必要：否

Vpc

指出執行個體是否位在 Virtual Private Cloud (VPC) 中。

類型：布林值

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

Parameter

服務: Amazon DocumentDB (with MongoDB compatibility)

有關個別參數的詳細資訊。

目錄

Note

在下列清單中，會先說明必要的參數。

AllowedValues

指定參數值的有效範圍。

類型：字串

必要：否

ApplyMethod

指出何時套用參數更新。

類型：字串

有效值: `immediate` | `pending-reboot`

必要：否

ApplyType

指定引擎特定的參數類型。

類型：字串

必要：否

DataType

指定參數的有效資料類型。

類型：字串

必要：否

Description

提供參數的描述。

類型：字串

必要：否

IsModifiable

指出是 (true) 否 (false) 可以修改參數。有些參數具有安全或操作上的隱含式，防止他們遭到變更。

類型：布林值

必要：否

MinimumEngineVersion

可套用參數的最早引擎版本。

類型：字串

必要：否

ParameterName

指定參數的名稱。

類型：字串

必要：否

ParameterValue

指定參數的值。

類型：字串

必要：否

Source

指出參數值的來源。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

PendingCloudwatchLogsExports

服務: Amazon DocumentDB (with MongoDB compatibility)

組態仍待處理的日誌類型清單。正在啟動的過程日誌正在運行的過程日誌。

內容

Note

在下列清單中，會先說明必要的參數。

LogTypesToDisable.member.N

正在進行啟用的日誌類型。啟用它們之後，這些日誌類型就會匯出到 Amazon CloudWatch 日誌。

類型：字串陣列

必要：否

LogTypesToEnable.member.N

正在進行停用的日誌類型。停用後，這些記錄類型就不會匯出至 CloudWatch 記錄檔。

類型：字串陣列

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

PendingMaintenanceAction

服務: Amazon DocumentDB (with MongoDB compatibility)

提供資源待處理維護動作的資訊。

內容

Note

在下列清單中，會先說明必要的參數。

Action

可供資源使用的待處理維護動作類型。

類型：字串

必要：否

AutoAppliedAfterDate

套用動作時的維護時段日期。維護動作會在此日期後的第一個維護時段中，套用到資源。如已指定此日期，則忽略任何 `next-maintenance` 選擇使用請求。

類型：Timestamp

必要：否

CurrentApplyDate

待處理維護動作套用至資源時的有效日期。

類型：Timestamp

必要：否

Description

提供維護動作更多詳細資訊的描述。

類型：字串

必要：否

ForcedApplyDate

自動套用維護動作時的日期。維護動作會在此日期套用至資源，無論資源的維護時段為何。如已指定此日期，則忽略任何 `immediate` 選擇使用請求。

類型：Timestamp

必要：否

OptInStatus

指出資源已收到的選擇使用請求類型。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

PendingModifiedValues

服務: Amazon DocumentDB (with MongoDB compatibility)

執行個體的一或多個修改過的設定。已要求這些修改過的設定，但尚未套用。

內容

Note

在下列清單中，會先說明必要的參數。

AllocatedStorage

包含將要套用或目前正在套用的執行個體的新AllocatedStorage大小。

類型：整數

必要：否

BackupRetentionPeriod

指定保留自動化備份的待處理天數。

類型：整數

必要：否

CACertificateIdentifier

指定資料庫執行個體的憑證授權單位 (CA) 憑證。

類型：字串

必要：否

DBInstanceClass

包含將要套用或目前正在套用的執行個體的新DBInstanceClass執行個體。

類型：字串

必要：否

DBInstanceIdentifier

包含將要套用或目前正在套用的執行個體的新DBInstanceIdentifier執行個體。

類型：字串

必要：否

DBSubnetGroupName

執行個體的新子網路群組。

類型：字串

必要：否

EngineVersion

指示資料庫引擎版本。

類型：字串

必要：否

Iops

為將套用或目前套用的執行個體指定新的佈建 IOPS 值。

類型：整數

必要：否

LicenseModel

執行個體的授權模式。

有效值：`license-included`、`bring-your-own-license`、`general-public-license`

類型：字串

必要：否

MasterUserPassword

包含執行處理之主要證明資料的擱置中或目前進行中的變更。

類型：字串

必要：否

MultiAZ

指示單一同步備份執行個體將變更為異地同步備份部署。

類型：布林值

必要：否

PendingCloudwatchLogsExports

組態仍待處理的日誌類型清單。這些日誌類型正在啟用或停用的過程中。

類型：[PendingCloudwatchLogsExports](#) 物件

必要：否

Port

指定執行個體的擱置中連接埠。

類型：整數

必要：否

StorageType

指定要和執行個體建立關聯的儲存體類型。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

ResourcePendingMaintenanceActions

服務: Amazon DocumentDB (with MongoDB compatibility)

代表的輸出[ApplyPendingMaintenanceAction](#)。

內容

Note

在下列清單中，會先說明必要的參數。

PendingMaintenanceActionDetails.PendingMaintenanceAction.N

清單，提供資源待處理維護動作的詳細資訊。

類型：[PendingMaintenanceAction](#) 物件陣列

必要：否

ResourceIdentifier

具有擱置的 Amazon Resource Name (ARN)。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

Subnet

服務: Amazon DocumentDB (with MongoDB compatibility)

子網路適用的相關詳資訊

內容

Note

在下列清單中，會先說明必要的參數。

SubnetAvailabilityZone

會指定子網路適用的可用區域。

類型：[AvailabilityZone](#) 物件

必要：否

SubnetIdentifier

指定子網路的識別符。

類型：字串

必要：否

SubnetStatus

指定子網路的狀態。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

Tag

服務: Amazon DocumentDB (with MongoDB compatibility)

指派給由金鑰值組成的 Amazon DocumentDB 資源的中繼資料。

內容

Note

在下列清單中，會先說明必要的參數。

Key

標籤必要的鍵名稱。字串值長度可以是 1 到 128 個 Unicode 字元，不可在前面加上aws:「"或rds:"。該字串僅能包含一組 Unicode 字母、數字、空格、「_」、「.」、「/」、「=」、「+」、「-」(Java regex : `^(?=[LZPN_!+=-])*$`)。

類型：字串

必要：否

Value

標籤的選用值。字串值長度可以是 1 到 256 個 Unicode 字元，不可在前面加上aws:「"或rds:"。該字串僅能包含一組 Unicode 字母、數字、空格、「_」、「.」、「/」、「=」、「+」、「-」(Java regex : `^(?=[LZPN_!+=-])*$`)。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

UpgradeTarget

服務: Amazon DocumentDB (with MongoDB compatibility)

執行個體可升級至的資料庫引擎版本。

內容

Note

在下列清單中，會先說明必要的參數。

AutoUpgrade

指出目標版本是否已套用至任何AutoMinorVersionUpgrade設定為的來源資料庫執行個體的值true。

類型：布林值

必要：否

Description

執行個體可升級至的資料庫引擎版本。

類型：字串

必要：否

Engine

升級目標資料庫引擎名稱。

類型：字串

必要：否

EngineVersion

升級目標資料庫引擎的版本編號。

類型：字串

必要：否

IsMajorVersionUpgrade

值，指出資料庫引擎是否升級到主要版本。

類型：布林值

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

VpcSecurityGroupMembership

服務: Amazon DocumentDB (with MongoDB compatibility)

用於 Virtual Private Cloud (VPC) 安全群組成員資格的回應元素。

內容

Note

在下列清單中，會先說明必要的參數。

Status

VPC 安全群組的狀態。

類型：字串

必要：否

VpcSecurityGroupId

VPC 安全群組的名稱。

類型：字串

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

Amazon DocumentDB Elastic Clusters

下列資料類型受 Amazon DocumentDB 彈性叢集支援：

- [Cluster](#)
- [ClusterInList](#)
- [ClusterSnapshot](#)
- [ClusterSnapshotInList](#)
- [Shard](#)
- [ValidationExceptionField](#)

Cluster

服務：Amazon DocumentDB Elastic Clusters

傳回特定彈性叢集的相關資訊。

目錄

Note

在下列清單中，會先說明必要的參數。

adminUserName

彈性叢集管理員的名稱。

類型：字串

必要：是

authType

彈性叢集的驗證類型。

類型：字串

有效值:PLAIN_TEXT | SECRET_ARN

必要：是

clusterArn

彈性叢集的 ARN 識別碼。

類型：字串

必要：是

clusterEndpoint

用來連線至彈性叢集的 URL。

類型：字串

必要：是

clusterName

彈性叢集的名稱。

類型：字串

必要：是

createTime

以世界協調時間 (UTC) 建立彈性叢集的時間。

類型：字串

必要：是

kmsKeyId

用來加密彈性叢集的 KMS 金鑰識別碼。

類型：字串

必要：是

preferredMaintenanceWindow

每週可能進行系統維護的時段，以國際標準時間 (UTC) 表示。

格式：ddd:hh24:mi-ddd:hh24:mi

類型：字串

必要：是

shardCapacity

指派給每個彈性叢集碎片的 vCPUs 數目。最大值為 64 個。允許的值為 2、4、8、16、32、64。

類型：整數

必要：是

shardCount

指派給彈性叢集的碎片數目。最大值為 32。

類型：整數

必要：是

status

彈性叢集的狀態。

類型：字串

有效值:CREATING | ACTIVE | DELETING | UPDATING |
VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED
| INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID |
INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN |
INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING |
SPLITTING | COPYING | STARTING | STOPPING | STOPPED

必要：是

subnetIds

彈性叢集的 Amazon EC2 子網路 ID。

類型：字串陣列

必要：是

vpcSecurityGroupIds

與此彈性叢集相關聯的 EC2 VPC 安全群組清單。

類型：字串陣列

必要：是

backupRetentionPeriod

自動快照的保留天數。

類型：整數

必要：否

preferredBackupWindow

如果啟用自動備份，則會建立自動備份的每日時間範圍 (由決定) backupRetentionPeriod。

類型：字串

必要：否

shardInstanceCount

套用至叢集中所有碎片的複本執行個體數目。shardInstanceCount值 1 表示有一個寫入器執行個體，而任何其他執行個體都是可用於讀取和改善可用性的複本。

類型：整數

必要：否

shards

叢集中的碎片總數。

類型：[Shard](#) 物件陣列

必要：否

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

ClusterInList

服務：Amazon DocumentDB Elastic Clusters

Amazon DocumentDB 彈性集群的列表。

目錄

Note

在下列清單中，會先說明必要的參數。

clusterArn

彈性叢集的 ARN 識別碼。

類型：字串

必要：是

clusterName

彈性叢集的名稱。

類型：字串

必要：是

status

彈性叢集的狀態。

類型：字串

有效值:CREATING | ACTIVE | DELETING | UPDATING |
VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED
| INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID |
INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN |
INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING |
SPLITTING | COPYING | STARTING | STOPPING | STOPPED

必要：是

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

ClusterSnapshot

服務：Amazon DocumentDB Elastic Clusters

傳回特定彈性叢集快照的相關資訊。

目錄

Note

在下列清單中，會先說明必要的參數。

adminUserName

彈性叢集管理員的名稱。

類型：字串

必要：是

clusterArn

彈性叢集的 ARN 識別碼。

類型：字串

必要：是

clusterCreationTime

以世界協調時間 (UTC) 建立彈性叢集的時間。

類型：字串

必要：是

kmsKeyId

KMS 金鑰識別符是 KMS 加密金鑰的 Amazon Resource Name (ARN)。如果您使用擁有此 KMS 加密金鑰的相同 Amazon 帳戶建立叢集，則可以使用 KMS 金鑰別名而非 ARN 做為 KMS 加密金鑰。如果未在此處指定加密金鑰，Amazon DocumentDB 會使用 KMS 為您的帳戶建立的預設加密金鑰。您的帳戶對每個 Amazon 區域都有不同的預設加密金鑰。

類型：字串

必要：是

snapshotArn

彈性叢集快照集的 ARN 識別碼。

類型：字串

必要：是

snapshotCreationTime

以世界協調時間 (UTC) 建立彈性叢集快照的時間。

類型：字串

必要：是

snapshotName

彈性叢集快照的名稱。

類型：字串

必要：是

status

彈性叢集快照的狀態。

類型：字串

有效值:CREATING | ACTIVE | DELETING | UPDATING |
VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED
| INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID |
INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN |
INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING |
SPLITTING | COPYING | STARTING | STOPPING | STOPPED

必要：是

subnetIds

彈性叢集的 Amazon EC2 子網路 ID。

類型：字串陣列

必要：是

vpcSecurityGroupIds

要與彈性叢集關聯的 EC2 VPC 安全群組清單。

類型：字串陣列

必要：是

snapshotType

要傳回的叢集快照類型。您可以指定下列其中一個值：

- automated-傳回 Amazon DocumentDB 為您 AWS 的帳戶自動建立的所有叢集快照。
- manual-傳回您為 AWS 帳戶手動建立的所有叢集快照。

類型：字串

有效值:MANUAL | AUTOMATED

必要：否

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

ClusterSnapshotInList

服務：Amazon DocumentDB Elastic Clusters

彈性叢集快照的清單。

目錄

Note

在下列清單中，會先說明必要的參數。

clusterArn

彈性叢集的 ARN 識別碼。

類型：字串

必要：是

snapshotArn

彈性叢集快照集的 ARN 識別碼。

類型：字串

必要：是

snapshotCreationTime

以世界協調時間 (UTC) 建立彈性叢集快照的時間。

類型：字串

必要：是

snapshotName

彈性叢集快照的名稱。

類型：字串

必要：是

status

彈性叢集快照的狀態。

類型：字串

有效值:CREATING | ACTIVE | DELETING | UPDATING |
VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED
| INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID |
INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN |
INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING |
SPLITTING | COPYING | STARTING | STOPPING | STOPPED

必要：是

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

Shard

服務：Amazon DocumentDB Elastic Clusters

碎片的名稱。

目錄

Note

在下列清單中，會先說明必要的參數。

`createTime`

以世界協調時間 (UTC) 建立碎片的時間。

類型：字串

必要：是

`shardId`

碎片的識別碼。

類型：字串

必要：是

`status`

碎片的當前狀態。

類型：字串

有效值:CREATING | ACTIVE | DELETING | UPDATING |
VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED
| INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID |
INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN |
INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING |
SPLITTING | COPYING | STARTING | STOPPING | STOPPED

必要：是

另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的軟件](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

ValidationExceptionField

服務: Amazon DocumentDB Elastic Clusters

發生指定驗證例外狀況的特定欄位。

內容

Note

在下列清單中，會先說明必要的參數。

message

此欄位中描述驗證例外狀況的錯誤訊息。

類型：字串

必要：是

name

發生驗證例外狀況的欄位名稱。

類型：字串

必要：是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

常見錯誤

本部分列出所有 AWS 服務 API 動作的常見錯誤。如需此服務之 API 動作的特定錯誤，請參閱該 API 動作的主題。

AccessDeniedException

您沒有足夠存取權可執行此動作。

HTTP 狀態碼：400

IncompleteSignature

請求簽署不符合 AWS 標準。

HTTP 狀態碼：400

InternalFailure

由於不明的錯誤、例外狀況或故障，處理請求失敗。

HTTP 狀態碼：500

InvalidAction

請求的動作或操作無效。確認已正確輸入動作。

HTTP 狀態碼：400

InvalidClientId

提供的 X.509 憑證或 AWS 存取金鑰 ID 不存在於我們的記錄中。

HTTP 狀態碼：403

NotAuthorized

您沒有執行此動作的許可。

HTTP 狀態碼：400

OptInRequired

AWS 存取金鑰 ID 需要訂閱服務。

HTTP 狀態碼：403

RequestExpired

請求送達服務已超過戳印日期於請求上之後的 15 分鐘，或者已超過請求過期日期之後的 15 分鐘 (例如預先簽章的 URL)，或者請求上的日期戳印在未來將超過 15 分鐘。

HTTP 狀態碼：400

ServiceUnavailable

由於伺服器暫時故障，請求失敗。

HTTP 狀態碼：503

ThrottlingException

由於請求調節，因此請求遭到拒絕。

HTTP 狀態碼：400

ValidationError

輸入不符合 AWS 服務規定的限制。

HTTP 狀態碼：400

常見參數

以下清單內含所有動作用來簽署 Signature 第 4 版請求的參數以及查詢字串。任何專屬於特定動作的參數則列於該動作的主題中。如需簽名版本 4 的詳細資訊，請參閱 IAM 使用者指南中的[簽署 AWS API 請求](#)。

Action

要執行的動作。

類型：字串

必要：是

Version

編寫請求所憑藉的 API 版本，以 YYYY-MM-DD 格式表示。

類型：字串

必要：是

X-Amz-Algorithm

建立請求簽章時所使用的雜湊演算法。

條件：當您在查詢字串中而非 HTTP 授權標頭中納入驗證資訊時，應指定此參數。

類型：字串

有效值: AWS4-HMAC-SHA256

必要：有條件

X-Amz-Credential

憑證範圍值，此為一個字串，其中包含您的存取金鑰、日期、您的目標區域、您請求的服務，以及終止字串 (“aws4_request”)。值以下列格式表示：access_key/YYYYMMDD/region/service/aws4_request。

如需詳細資訊，請參閱 IAM 使用者指南中的 [建立已簽署的AWS API 請求](#)。

條件：當您在查詢字串中而非 HTTP 授權標頭中納入驗證資訊時，應指定此參數。

類型：字串

必要：有條件

X-Amz-Date

用來建立簽署的日期。格式必須是 ISO 8601 基本格式 (YYYYMMDD'T'HHMMSS'Z')。例如，以下日期時間是有效的 X-Amz-Date 值：20120325T120000Z

條件：對所有請求而言，X-Amz-Date 皆為選用，可用來覆寫用於簽署請求的日期。如果規定日期標頭採用 ISO 8601 基本格式，則不需要 X-Amz-Date。當使用 X-Amz-Date 時，其一律會覆寫日期標頭的值。如需詳細資訊，請參閱 IAM 使用者指南中的 [AWSAPI 請求簽名元素](#)。

類型：字串

必要：有條件

X-Amz-Security-Token

透過呼叫AWS Security Token Service (AWS STS) 所取得的臨時安全字符。如需支援 IAM 使用者指南中的臨時安全憑證的服務清單AWS STS [AWS 服務](#)，請前往 IAM 使用者指南中的《[可搭配 IAM 運作](#)》。

條件：如果您使用安全憑證AWS STS，則必須納入安全字符。

類型：字串

必要：有條件

X-Amz-Signature

指定從要簽署的字串和衍生的簽署金鑰中計算出的十六進位編碼簽章。

條件：當您在查詢字串中而非 HTTP 授權標頭中納入驗證資訊時，應指定此參數。

類型：字串

必要：有條件

X-Amz-SignedHeaders

指定納入作為標準請求一部分的所有 HTTP 標頭。如需有關指定已簽署標頭的詳細資訊，請參閱 IAM 使用者指南中的[建立已簽署AWS API 請求](#)。

條件：當您在查詢字串中而非 HTTP 授權標頭中納入驗證資訊時，應指定此參數。

類型：字串

必要：有條件

版本備註

這些版本說明按發行日期說明 Amazon DocumentDB 的功能、改進功能和錯誤修正。發行說明包括所有 Amazon DocumentDB 引擎版本的更新。

您可以執行下列命令來判斷目前的 Amazon DocumentDB 引擎修補程式版本：

```
db.runCommand({getEngineVersion: 1})
```

如果您的叢集不在最新版本的引擎上，很可能是您有擱置中的維護可用來升級您的引擎。如需詳細資訊，請參閱[維護 Amazon DocumentDB](#)開發人員指南中的。

主題

- [2024年4月3日](#)
- [2024年2月22日](#)
- [2024年1月30日](#)
- [2024年1月10日](#)
- [2023年12月20日](#)
- [2023年12月13日](#)
- [2023年11月29日](#)
- [2023年11月21日](#)
- [2023年11月17日](#)
- [2023年11月6日](#)
- [2023年10月20日](#)
- [2023年9月25日](#)
- [2023年9月20日](#)
- [2023年9月15日](#)
- [2023年9月11日](#)
- [2023年8月3日](#)
- [2023年7月13日](#)
- [2023年6月7日](#)

- [2023 年 5 月 10 日](#)
- [2023 年 4 月 4 日](#)
- [2023 年 3 月 22 日](#)
- [2023 年 3 月 1 日](#)
- [2023 年 2 月 27 日](#)
- [2023 年 2 月 2 日](#)
- [2022 年 11 月 30 日](#)
- [2022 年 8 月 9 日](#)
- [2022 年 7 月 25 日](#)
- [2022 年 6 月 27 日](#)
- [2022 年 4 月 29 日](#)
- [2022 年 4 月 7 日](#)
- [2022 年 3 月 16 日](#)
- [2022 年 2 月 8 日](#)
- [2022 年 1 月 24 日](#)
- [2022 年 1 月 21 日](#)
- [2021 年 10 月 25 日](#)
- [2021 年 6 月 24 日](#)
- [2021 年 5 月 4 日](#)
- [2021 年 1 月 15 日](#)
- [2020 年 11 月 9 日](#)
- [2020 年 10 月 30 日](#)
- [2020 年 9 月 22 日](#)
- [2020 年 7 月 10 日](#)
- [2020 年 6 月 30 日](#)

2024年4月3日

Amazon DocumentDB 現已在中東 (阿拉伯聯合大公國) 區域提供。如需詳細資訊，請參閱此[部落格文章](#)。

Note

下列 Amazon DocumentDB 引擎修補程式正在接下來的幾週內交付到所有 Amazon DocumentDB 區域。當此引擎修補程式在您所在地區可用時，您將透過 AWS Health 儀表板 (AHD) 收到服務修補程式通知，AWS Management Console 並透過電子郵件傳送至您 AWS 帳戶的 root 使用者電子郵件地址。

此引擎修補程式包含下列新功能和錯誤修正。請注意，當引擎修補程式在所有地區推出後，下列清單及相關的支援文件可能會更新為包含其他功能公告。

新功能

Amazon DocumentDB 5.0 (引擎修補程序版本 3.0.5721)

- 增bypassDocumentValidation加了對\$jsonSchema.
- 增加了支持\$expr.
- 在中新增了對不相關聯結的支援。\$lookup
- 新增在\$out彙總階段保留驗證規則的支援。

Amazon DocumentDB 4.0 (引擎修補程式版本 2.0.10392)

- 增bypassDocumentValidation加了對\$jsonSchema.
- 增加了支持\$expr.
- 在中新增了對不相關聯結的支援。\$lookup
- 新增在\$out彙總階段保留驗證規則的支援。

錯誤修復和其他更改

- 修復了db.coll.stats()在蒙戈殼 1.7 版及更高版本上調用時的錯誤。
- 修正包含\$regex為相同彙總管線一部分的變更串流查詢的記憶體洩漏問題。

2024年2月22 日

新功能

Amazon DocumentDB 彈性集群

Amazon DocumentDB 彈性叢集現在支援下列功能：

- 可讀取的次要碎片執行個體複本-如需詳細資訊，請參閱 [步驟 1：建立彈性叢集](#)
- 啟動/停止群集-如需詳細資訊，請參閱 [停止和啟動 Amazon DocumentDB 彈性叢集](#)。
- 可設定的碎片執行個體-如需詳細資訊，請參閱的步驟 5b。 [步驟 1：建立彈性叢集](#)
- 快照的自動備份-如需詳細資訊，請參閱 [管理彈性叢集快照自動備份](#)。
- 複製快照-如需詳細資訊，請參閱 [複製彈性叢集快照](#)。

2024年1月30日

新功能

Amazon DocumentDB 彈性集群

Amazon DocumentDB 彈性叢集現在可在下列區域使用：

- 亞太區域 (孟買)
- 亞太區域 (首爾)
- 南美洲 (聖保羅)
- 歐洲 (倫敦)

如需詳細資訊，請參閱 [彈性叢集區域和版本可用性](#)。

Amazon DocumentDB 全球集群

全域叢集現在可在這兩個 AWS GovCloud (US) 區域使用：AWS GovCloud (美國東部) 和 AWS GovCloud (美國西部)。

2024 年 1 月 10 日

新功能

Amazon DocumentDB 5.0 (引擎修補程式版本)

- 增加了對 HNSW 向量索引的支援。如需詳細資訊，請參閱 [向量搜索 Amazon DocumentDB](#)。
- 增加了對部分索引的支持。如需詳細資訊，請參閱 [部分索引](#)。
- 已新增對currentOp指令內集合上 GC 執行階段的支援。
- 在 Amazon DocumentDB 上添加了對本地文本搜索的文本索引支持。如需詳細資訊，請參閱 [使用 Amazon DocumentDB 執行文本搜索](#)。
- 已新增\$jsonSchema架構關鍵字typeallOf、oneOf、anyOf、not、maxItems、minItems、maxProperties、minProperties和的支援uniqueItems。
如需更多資訊，請參閱[使用 JSON 結構定義驗證](#)。
- 增加了對算術運算子\$ceil\$floor、\$ln、\$log、\$log10\$sqrt、和的支援\$exp。
如需更多資訊，請參閱[算術運算子](#)。
- 增加了對 parallel IVFFLAT 向量索引構建的支援。通過從開發人員指南中刪除 parallel IVFFLAT 向量索引構建限制來更新文檔。

Amazon DocumentDB 4.0 (引擎修補程式版本)

- 已新增對currentOp指令內集合上 GC 執行階段的支援。
- 已新增\$jsonSchema架構關鍵字typeallOf、oneOf、anyOf、not、maxItems、minItems、maxProperties、minProperties和的支援uniqueItems。
如需更多資訊，請參閱[使用 JSON 結構定義驗證](#)。
- 增加了對算術運算子\$ceil\$floor、\$ln、\$log、\$log10\$sqrt、和的支援\$exp。
如需更多資訊，請參閱[算術運算子](#)。

錯誤修復和其他更改

- 為調用添加了不區分大小寫的功能。db.runCommand("dbstats")使用 3.0.4960 或 2.0.10221 之前版本的引擎修補程式版本的 Amazon DocumentDB 5.0 和 4.0 客戶應套用這些最新的引擎修補程式。
- 修復了db.coll.stats()在蒙戈殼 1.7 版及更高版本上調用時的錯誤。通過從開發人員指南中刪除 mongo shell db.coll.stats() 故障排除提示來更新文檔。

2023 年 12 月 20 日

其他變更

在 Amazon DocumentDB 3.6 和 4.0 中啟用了就地主要版本升級的支援。如需詳細資訊，請參閱 [Amazon DocumentDB 就地主要版本升級](#)。

2023 年 12 月 13 日

新功能

新增對一鍵式 EC2 連線的支援。如需詳細資訊，請參閱 [使用 Amazon EC2 Connect](#)。

2023 年 11 月 29 日

Amazon DocumentDB 5.0 (引擎修補程式版本 3.0.3727)

新功能

增加了對向量搜索的支持。如需詳細資訊，請參閱此[部落格文章](#)，並造訪 Amazon DocumentDB 開發人員指南[向量搜索 Amazon DocumentDB](#)中的。

2023 年 11 月 21 日

Amazon DocumentDB 5.0 (引擎修補程式版本 3.0.3727)

新功能

增加了對 I/O 優化存儲的支持。如需詳細資訊，請參閱 Amazon DocumentDB 開發人員指南[Amazon DocumentDB 群集存儲配置](#)中的。

添加了與 SageMaker Canvas 無代碼機器學習的集成。如需詳細資訊，請參閱 Amazon DocumentDB 開發人員指南[使用 Amazon SageMaker 畫布進行無程式碼機器](#)中的。

2023 年 11 月 17 日

新功能

Amazon DocumentDB 現已在 AWS GovCloud (美國東部) 區域提供。如需詳細資訊，請參閱此[部落格文章](#)。

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.208570)

使用者定義的區域變數名稱現在支援投影運算子 (如`$let`和) 的「`_`」(底線) `$filter`。

2023 年 11 月 6 日

Amazon DocumentDB 5.0 (引擎修補程式版本 3.0.3727) 和 4.0 (引擎修補程式版本 2.0.9876)

新功能

- 增加了對`$jsonSchema`結構描述關鍵字`maxLength``minLength`、`maximum`、`minimum`、`exclusiveMaximum`、`exclusiveMinimum`和的支援`additionalItems`。

請注意，JSON 結構描述驗證僅支援執行個體型叢集。

- 已新增對`$convert`彙總管線運算子及其速記衍生運算子`$toBool`、`$toInt`、`$toLong`、`$toDouble``$toString``$toDecimal``$toObjectId`、和的支援。`$toDate`
- 增加了對集合運算式運算子`$setDifference``$anyElementTrue`、和的支援`$allElementTrue`。

錯誤修復和其他更改

修正未顯示從`-NaN`變更串流更新`NaN`為的問題。

2023 年 10 月 20 日

其他變更

Amazon DocumentDB 已經發現了一個問題，並暫時禁止在所有區域進行主要版本升級 (MVU)。我們已經確定了問題的根本原因，並開發了目前正在進行測試的修復程序。我們預計此修正程式將在 2023 年第 4 季結束前部署於所有區域。MVU 將保持停用狀態，直到修正程式部署到所有區域。如需 MVU 功能可用性的更多更新，請參閱此版本說明頁面。

在此期間，您可以使用 AWS DMS 將 Amazon DocumentDB 資料庫從較低版本的叢集遷移到更高版本來執行主要版本升級。請按照中的步驟[使用升級您的 Amazon DocumentDB 集群 AWS Database Migration Service](#)進行升級 AWS DMS。您也可以參考此[部落格文章](#)，瞭解使用升級時應遵循的最佳作法的其他資訊 AWS DMS。

2023 年 9 月 25 日

新功能

Amazon DocumentDB 現已在亞太區域 (香港) 區域提供。如需詳細資訊，請參閱此[部落格文章](#)。

2023 年 9 月 20 日

新功能

在 Amazon DocumentDB 3.6 和 4.0 中添加了對就地主要版本升級的支持。如需更多資訊，請參閱[Amazon DocumentDB 就地主要版本升級](#)。

2023 年 9 月 15 日

新功能

Amazon DocumentDB 5.0 (引擎修補程式版本 3.0.3140) 和 4.0 (引擎修補程式版本 2.0.9686)

- 僅在執行個體型叢集中新增對 \$JsonSchema 結構描述驗證程式的支援。

如需更多資訊，請參閱[使用 JSON 結構定義驗證](#)。

2023 年 9 月 11 日

新功能

亞太區域 (海德拉巴) 區域目前提供亞 Amazon DocumentDB。如需詳細資訊，請參閱此[部落格文章](#)。

2023 年 8 月 3 日

新功能

Amazon DocumentDB 彈性集群

- Amazon DocumentDB 彈性叢集現在支援下列操作：
 - top
 - collStats
 - hint
 - dataSize

[支援的 MongoDB API、營運和資料類型](#)如需支援指令和作業的完整清單，請參閱。

- 現在支援存留時間 (TTL) 索引。
- 索引hints現在支援索引運算式。

2023 年 7 月 13 日

新功能

Amazon DocumentDB 5.0 (引擎修補程式版本 3.0.1948)

- 增加了對文檔壓縮的支持。
- 增加了對 parallel 索引構建的支持。
- 增加了對索引構建狀態的支持。

Amazon DocumentDB 4.0 (引擎修補程序版本 2.0.9259)

- 增加了對 parallel 索引構建的支持。

錯誤修復和其他更改

Amazon DocumentDB 5.0 (引擎修補程式版本 3.0.1948)

- 修正使用者無法存取系統集createCollection合時，Amazon DocumentDB 彈性叢集的身份驗證問題。
- 修正次要區域執行個體無法使用相同主要區域執行個體名稱的問題。

Amazon DocumentDB 4.0 (引擎修補程序版本 2.0.9259)

- 停止將內部監視查詢新增至稽核記錄檔。

2023 年 6 月 7 日

錯誤修復和其他更改

Amazon DocumentDB 5.0

- 現在，Amazon DocumentDB 5.0 支持 r5 和 t3.medium 實例。
- engineVersion選項預設值5.0.0在 AWS SDK AWS CLI、和中 AWS CloudFormation。

2023 年 5 月 10 日

錯誤修復和其他更改

Amazon DocumentDB 5.0 (引擎修補程序版本 3.0.1361)

- 在createIndex指令中增加ignoreunknownindexoptions了對的支援。
- 停止將內部監視查詢新增至稽核記錄檔。
- 使用者定義的區域變數名稱現在支援投影運算子 (如\$let和) 的「_」(底線) \$filter。

2023 年 4 月 4 日

錯誤修復和其他更改

Amazon DocumentDB 4.0 (引擎修補程式版本 2.0.8934)

- 已修正在進行的工作負載期間啟用 DML 稽核的問題。
- 已修正當含有提示的彙總命令傳遞字串值時，DML 稽核的問題。
- 修正了當具有兩個授權集合和僅名稱選項設置為 true 的讀寫任何數據庫角色的用戶 `listCollections` 命令不起作用的問題。
- 修正正確剖析欄位名稱中數值字串的問題。
- 當它們影響垃圾回收時，取消長時間運行的游標。
- 使用者定義的區域變數名稱現在支援投影運算子 (如 `$let` 和) 的「_」(底線) `$filter`。

2023 年 3 月 22 日

新功能

Amazon DocumentDB 彈性叢集現已在亞太區域 (新加坡)、亞太區域 (雪梨) 和亞太區域 (東京) 區域推出。如需詳細資訊，請參閱 [彈性叢集區域和版本可用性](#)。

2023 年 3 月 1 日

新功能

Amazon DocumentDB 5.0 (引擎修補程序版本 3.0.775)

- 引入 Amazon DocumentDB 5.0
 - 兼容性 (支持 MongoDB 5.0 API 驅動程序)
 - Support 用戶端欄位層級加密 (FLE)。您現在可以在將資料寫入 Amazon DocumentDB 叢集之前，先在用戶端加密欄位。如需詳細資訊，請參閱 [用戶端欄位層級加密](#)
 - 新的聚合運算符：`$dateAdd`，`$dateSubtract`
- 對於所有以執行個體為基礎的 Amazon DocumentDB 叢集和以資料為基礎的彈性叢集，將儲存限制增加到 128 TiB。
- Amazon DocumentDB 5.0 現在支援在第一個巢狀層級的 `$elemMatch` 操作員進行索引掃描。如果查詢只有一個層級的 `$elemMatch` 篩選器，且巢狀 `$elemMatch` 查詢不支援索引掃描，則支援索引掃描。

支援索引掃描的查詢形狀：

```
db.foo.find( { "a": { $elemMatch: { "b": "xyz", "c": "abc" } } })
```


不支援索引掃描的查詢形狀：

```
db.foo.find( { "a": { $elemMatch: { "b": { $elemMatch: { "d": "xyz", "e": "abc" } } } } })
```

2023 年 2 月 27 日

錯誤修復和其他更改

Amazon DocumentDB 4.0

增加了對 AWS Lambda. 如需詳細資訊，請參閱[AWS Lambda 搭配變更串流使用](#)。

2023 年 2 月 2 日

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.208432)

- 已修正在進行的工作負載期間啟用 DML 稽核的問題。
- 已修正當含有提示的彙總命令傳遞字串值時，DML 稽核的問題。
- 修正了當具有兩個授權集合和僅名稱選項設置為 true 的讀寫任何數據庫角色的用戶 listCollections 命令不起作用的問題。
- 修正正確剖析欄位名稱中數值字串的問題。
- 當它們影響垃圾回收時，取消長時間運行的游標。

2022 年 11 月 30 日

新功能

Amazon DocumentDB 彈性集群

Amazon DocumentDB 彈性叢集是一種新類型的 Amazon DocumentDB 叢集，可讓使用者利用 MongoDB 資料分割 API 向外擴展其叢集。彈性叢集可在多個基礎運算執行個體和磁碟區之間分配資料和運算，以 PB 級的儲存容量處理幾乎任何數量的讀取和寫入。若要進一步了解，請參閱[使用 Amazon DocumentDB 彈性叢集](#)。

2022 年 8 月 9 日

新功能

Amazon DocumentDB 3.6 (引擎修補程式版本) 和 4.0

- 增加了對十進制 128 數據類型的支援。十進制 128 是在所有地區的文檔數據庫是可行的支援 BSON 數據類型。

如需詳細資訊，請參閱[資料類型](#)。

- 增加了對使用 Amazon CloudWatch 日誌進行 DML 查詢審核的支援。現在，Amazon DocumentDB 可以將資料操縱語言 (DML) 事件和資料定義語言 (DDL) 事件記錄到 Amazon 日誌中。CloudWatch

如需詳細資訊，請參閱此[部落格文章](#)。

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本) 和 4.0

- 現在，您可以使用具有權限的密碼更改自己的密碼。changeOwnPassword

2022 年 7 月 25 日

新功能

Amazon DocumentDB 4.0

您現在可以利用建立使用相同 DocumentDB 叢集磁碟區並具有與原始叢集相同資料的複製，更快速地建立叢集。如需詳細資訊，請參閱[管理 Amazon DocumentDB 叢集](#)。

2022 年 6 月 27 日

新功能

Amazon DocumentDB 4.0 (引擎修補程序版本 2.0.7509)

Amazon DocumentDB 會根據使用模式動態調整資料庫大小。新增更多資料可增加高達 64 TB (TiB) 的空間，而刪除資料則會減少配置的空間。

2022 年 4 月 29 日

新功能

Amazon DocumentDB 現已在中國（北京）地區提供。如需詳細資訊，請參閱此[部落格文章](#)。

2022 年 4 月 7 日

新功能

Amazon DocumentDB 3.6 (引擎修補程式版本) 和 4.0 (引擎修補程式版本 2.0.6142 和 2.0.6948 版)

Amazon DocumentDB Performance Insights 現已提供預覽版。您現在可以在滾動式視窗中儲存七天的效能歷史記錄，而無需額外付費。如需詳細資訊，請參閱[使用 Performance Insights 進行監視](#)。

2022 年 3 月 16 日

新功能

Amazon DocumentDB 現已在歐洲（米蘭）地區提供。如需詳細資訊，請參閱此[部落格文章](#)。

2022 年 2 月 8 日

新功能

亞 Amazon DocumentDB R6g 和 T4G 執行個體現已在亞太地區、南美洲和歐洲提供。如需詳細資訊，請參閱此[部落格文章](#)。

2022 年 1 月 24 日

新功能

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.207684) 和 4.0 (引擎修補程式版本 2.0.5170)

- DocDB; 現在提供了一個免費試用。有關詳細信息，請參閱 [Amazon DocumentDB 免費試用頁面](#)。
- 您現在可以將增強功能與地理空間查詢搭配使用，包括以下 API：
 - \$geoWithin
 - \$geoIntersects

- 已新增對下列 MongoDB 運算子的支援：
 - `$mergeObjects`
 - `$reduce`

如需詳細資訊，請參閱[使用 Amazon DocumentDB 查詢地理空間資料](#)。

2022 年 1 月 21 日

新功能

Amazon DocumentDB 4.0 (引擎修補程式版本 2.0.5706)

- 現在支援 Amazon DocumentDB 重力量 2 (r6 g. 大型、r6 g .2 x 大、r6 g .4 x 大、r6 g.

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.207781) 和 4.0 (引擎修補程式版本 2.0.5706)

- 已新增對下列 MongoDB 應用程式介面的支援：
 - `$reduce`
 - `$mergeObjects`
 - `$geoWithin`
 - `$geoIntersects`

2021 年 10 月 25 日

新功能

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.207780) 和 4.0 (引擎修補程式版本 2.0.5704)

- 增加了對以下 MongoDB API 的支持
 - `$literal`
 - `$map`
 - `$$ROOT`
- Support 查 GeoSpatial 詢功能。有關更多詳細信息，請參閱[此博客](#)
- Support 使用者定義角色的存取控制。有關更多詳細信息，請參閱[此博客](#)

- Amazon DocumentDB JDBC 驅動程序，以啟用從 BI 工具，如 Tableau 和查詢工具，如 SQL 工作台的連接

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.207780) 和 4.0 (引擎修補程式版本 2.0.5704)

- 錯誤修復了當顯 `.sort()` 式出現時正確排序 `$natural $natural`
- 更改流的錯誤修復工作 `$redact`
- 使用空陣列的 `$ifNull` 錯誤修正
- 當目前登入的使用者遭到刪除，或撤銷該使用者正在進行中活動的權限時，過度耗用資源/伺服器當機的錯誤修正
- 錯誤修復 `listDatabase` 和 `listCollection` 權限檢查
- 錯誤修復了多鍵元素的去重複邏輯

2021 年 6 月 24 日

新功能

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.207117) 和 4.0 (引擎修補程式版本 2.0.3371)

- 現在支援 R5.8 大型執行個體和 R5.16 大型執行個體。若要進一步了解，請參閱部落格文章 [亞馬遜文件資料庫現在支援 R5.8 x 和 R5.16 大型執行個體](#)。
- 現在支援 [全域叢集](#)，可從區域範圍內的中斷提供災難復原，並允許從最近的 Amazon DocumentDB 叢集讀取，以啟用低延遲的全域讀取。

2021 年 5 月 4 日

新功能

查看此 [博客文章](#) 中的所有新功能。

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.207117) 和 4.0 (引擎修補程式版本 2.0.3371)

- `renameCollection`
- `$zip`

- `$indexOfArray`
- `$reverseArray`
- `$natural`
- `$hint` 支持更新
- 索引掃描 `distinct`

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本 1.0.207117) 和 4.0 (引擎修補程式版本 2.0.3371)

- 減少 `$in` 查詢的記憶體使用量
- 修復了多鍵索引中的內存洩漏
- 已修正下列項目的說明計畫和效能分析工具輸出 `$out`
- 為內部監控系統的操作添加了超時，以提高可靠性
- 修復了影響傳遞給多鍵索引的查詢謂詞的缺陷

2021 年 1 月 15 日

新功能

Amazon DocumentDB 4.0 (引擎修補程序版本 2.0.722)

- 無

Amazon DocumentDB 3.6 (引擎修補程式版本)

- 能夠將索引與 `$lookup` 聚合階段一起使用
- `find()` 具有預測的查詢可以從索引 (覆蓋的查詢) 提供方向
- 能夠 `hint()` 與 `findAndModify`
- 操作員的效能最佳化 `$addToSet`
- 減少整體索引大小的改進
- 新的彙總運算
子：`$ifNull$replaceRoot$setIsSubset`、`$setIntersection`、`$setUnion`、和
`$setEquals`

- 使用者也可以在KillCursor不需要角色的情況下結束自己的游標

2020 年 11 月 9 日

新功能

查看此[博客文章](#)中的所有新功能。

Amazon DocumentDB 4.0 (引擎修補程序版本 2.0.722)

- MongoDB 4.0 兼容性
- ACID 交易
- Support `cluster(client.watch()` 或 `mongo.watch()`) 和資料庫層級 (`db.watch()`) 變更串流
- 能夠使用以下方式啟動或繼續變更串流 `startAtOperationTime`
- 將變更串流保留期延長至 7 天 (之前 24 小時)
- AWS DMS Amazon DocumentDB 4.0 的目標
- CloudWatch 量度: `TransactionsOpenTransactionsOpenMaxTransactionsAborted`、`TransactionsStarted`、和 `TransactionsCommitted`
- `currentOp`、`ServerStatus` 和中交易的新欄位 `profiler`。
- 能夠將索引與 `$lookup` 聚合階段一起使用
- `find()` 具有預測的查詢可以從索引 (覆蓋的查詢) 提供方向
- 能夠 `hint()` 與 `findAndModify`
- 操作員的效能最佳化 `$addToSet`
- 減少整體索引大小的改進。
- 新的彙總運算子: `$ifNull$replaceRoot$setIsSubset`、`$setIntersection`、`$setUnion`、和 `$setEquals`
- 使用 `ListCollection` 和 `ListDatabase` 命令，您現在可以選擇性地使用 `authorizedCollections` 和 `authorizedDatabases` 參數，讓使用者分別列出他們有權存取的集合和資料庫，而不需要 `listCollections` 和 `listDatabase` 角色
- 使用者也可以在KillCursor不需要角色的情況下結束自己的游標
- 比較子文件的數值類型現在與比較第一層文件的數值類型一致。Amazon DocumentDB 4.0 中的行為現在與 MongoDB 兼容。

Amazon DocumentDB 3.6 (引擎修補程式版本)

- 無

錯誤修復和其他更改

Amazon DocumentDB 4.0 (引擎修補程序版本 2.0.722)

- `$setOnInsert`使用位置運算子`$`時，不再允許更新。Amazon DocumentDB 4.0 中的行為現在與 MongoDB 兼容。
- 固定的問題`$createCollection`和設置 `autoIndexId`
- 巢狀文件的投影
- 更改了工作內存的默認設置以隨實例內存大小進行擴展
- 垃圾收集改進
- 在路徑中使用空鍵查找，與 `mongo` 的行為差異
- 在時區行為`dateToString`修正錯誤
- 固定`$push` (聚合) 以尊重排序順序
- 修正了`$currentOp`聚合
- 固定的問題與`readPreference`二級
- 修復了驗證`$createIndex`問題與發出命令的數據庫相同
- 修正了不一致的行為`minKey`，`maxKey`查找失敗
- 修復了`$size`操作員不使用複合數組的問題
- 修復了正則表達式否定的問`$in`題
- 修復了對視圖運行`$distinct`命令的問題
- 修復了聚合和查找命令以不同方式排序缺少字段的問題
- 固定`$eq`到正則表達式不檢查類型
- `$currentDate`修正了時間戳序數位置行為中的錯誤
- 固定毫秒粒度 `$currentDate`

Amazon DocumentDB 3.6 (引擎修補程式版本)

- 無

2020 年 10 月 30 日

新功能

查看此[博客文章](#)中的所有新功能。

Amazon DocumentDB 3.6 (引擎修補程式版本)

- 增加了在群集級別(`client.watch()`或數據庫中打開更改流光標`mongo.watch()`)的功能 (`db.watch()`)
- 能夠將更改流保留期限增加到 7 天 (之前 24 小時)

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本)

- 各種一般案例效能改善
- 目標性的安全性改善
- 修正了一個複合索引的第二個字段跳過排序的問題
- 在多鍵索引 (不是複合) 的單個字段上啟用常規索引的相等
- 固定驗證競爭條件
- 修正造成不常發生垃圾回收當機的問題
- RBAC 安全性改善
- 已新增`databaseConnectionsMax`量度
- `r5.24xlarge`執行個體上的某些工作負載的效能

2020 年 9 月 22 日

新功能

查看此[博客文章](#)中的所有新功能。

Amazon DocumentDB 3.6 (引擎修補程式版本)

- `$out`彙總階段

- 將每個執行個體的最大連線和游標數目增加 10 倍

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本)

- 無

2020 年 7 月 10 日

新功能

查看此[博客文章](#)中的所有新功能。

Amazon DocumentDB 3.6 (引擎修補程式版本)

- 跨區域快照複製

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本)

- 無

2020 年 6 月 30 日

新功能

查看此[博客文章](#)中的所有新功能。

Amazon DocumentDB 3.6 (引擎修補程式版本)

- T3 中型執行個體

錯誤修復和其他更改

Amazon DocumentDB 3.6 (引擎修補程式版本)

- t3 執行個體的閒置記憶體回收
- 驗證改進
- 改善 SASL 驗證效能
- 修currentOp正超過最大可能操作時的問題
- 修killOps正了批量更新和刪除的問題
- 改善\$sample效能 \$match
- 修正了\$\$在編輯階段的情況下支持
- 修復了各種循環崩潰根本原因
- 改進 TTL 掃描以減少 iOS 和延遲
- 最佳化記憶體使用率 \$unwind
- 固定收集統計競爭條件下降索引
- 固定並發索引構建期間的競爭條件
- 修復了索引中hash_search的罕見崩潰

Amazon DocumentDB 開發人員指南的文檔歷史記錄

- API 版本：2014-10-31
- 最新文件更新：2023 年 6 月 2 日

下表說明此版本的 Amazon DocumentDB 開發人員指南的文件。

| 變更 | 描述 | 日期 |
|---------------------------------|--|------------------|
| AWS 受管策略更新-策略變更 | Amazon DocumentDB 更新彈性叢集的完整存取政策。 | 2024年2月21 日 |
| AWS 受管策略更新-策略變更 | Amazon DocumentDB 會更新彈性叢集的唯一讀和完整存取政策。 | 2023 年 6 月 21 日 |
| AWS 受管策略更新-新策略 | Amazon DocumentDB 為彈性叢集引入了新的唯一讀政策。 | 2023 年 6 月 8 日 |
| AWS 受管策略更新-新策略 | Amazon DocumentDB 為彈性叢集引入了新的完整存取政策。 | 2023 年 6 月 5 日 |
| MongoDB 性 | Amazon DocumentDB 現在與版本的 MongoDB 的 5.0 兼容。 | 2023 年 3 月 1 日 |
| 政策更新 | 為了支援 Amazon DocumentDB 彈性叢集功能，AmazonDoc 資料庫ConsoleFullAccess 政策已更新，並引入 AmazonDoc 資料庫。ElasticService RolePolicy | 2022 年 11 月 30 日 |
| 彈性叢集 | 新增彈性叢集功能，支援跨 Amazon DocumentDB 分散式 | 2022 年 11 月 30 日 |

| | | |
|---|---|-----------------|
| | 儲存系統的資料進行雜湊式分割 (分片)。 | |
| 全域叢集 | 已新增有關如何使用全域叢集的文件。 | 2021 年 6 月 2 日 |
| 活動訂閱 | 增加了事件訂閱文檔。 | 2021 年 3 月 26 日 |
| 3.6 版本升級 | 在角色型存取控制、彙總運算子和效能方面，對 3.6 版的改進已記錄。 | 2021 年 1 月 15 日 |
| MongoDB 4.0 兼容性 | Amazon DocumentDB 現在與版本 4.0 的 MongoDB 兼容。 | 2020 年 11 月 9 日 |
| 開始使用指南 | 使用亞馬遜 EC2 AWS Cloud9、Robo3T 或工作室 3T 開始使用 Amazon DocumentDB 的新入門指南。 | 2020年8月15日 |
| 支援其他可用區域 | 亞 Amazon DocumentDB 增加了對亞太區域 (首爾) (AP-東北 -2) 的其他可用區域的支持。 | 2020年7月14日 |
| 新增跨區域複製快照的支援。 | Amazon DocumentDB 增加了對跨集群快照複製的支持。AWS 區域如需詳細資訊，請參閱 跨區域複製快照 。 | 2020 年 7 月 10 日 |
| 增加了對 T3 實例類的支持。 | 已新增支援所有支援 Amazon DocumentDB 的區域中的 T3 執行個體類型。如需詳細資訊，請參閱 區域和執行個體類別規格支援的執行個體類別 。 | 2020 年 6 月 30 日 |
| 增加了對 AWS GovCloud (US). | Amazon DocumentDB 現已在該 AWS GovCloud (US) 地區 (us-gov-west-1) 提供。 | 2020 年 6 月 29 日 |

| | | |
|---|---|-----------------|
| 增加了 16 個新 CloudWatch 指標。 | Amazon DocumentDB 增加了對 16 個新 Amazon CloudWatch 指標的支持。如需詳細資訊， 請參閱使用 CloudWatch | 2020 年 6 月 23 日 |
| 增加了對空字符和 \$ 正則表達式運算符的支持。 | Amazon DocumentDB 增加了對字符串中的空字符的支持，並為 \$ 正則表達式使用索引的能力。若要檢視支援的 MongoDB API 和彙總管道功能，請參閱與 MongoAmazon DocumentDB 的 功能差異 。 | 2020 年 6 月 22 日 |
| 增加了對改進的多鍵索引功能的支持。 | Amazon DocumentDB 增加了對改進的多重金鑰索引功能的支援，其中包括對超過 2,048 位元組的陣列進行索引，以及在同一陣列中使用多個金鑰建立複合多鍵索引的功能。如需詳細資訊，請參閱 MongoDB 的功能差異 。 | 2020 年 4 月 23 日 |
| 增加了對 Amazon DocumentDB AWS CloudFormation 堆棧的刪除保護的支持。 | Amazon DocumentDB 增加了在創建 Amazon DocumentDB AWS CloudFormation 堆棧時啟用刪除保護的支持。 | 2020 年 4 月 20 日 |
| 增加了對基於角色的訪問控制的支持。 | Amazon DocumentDB 新增了使用內建角色對以角色為基礎的存取控制的支援。 | 2020 年 3 月 26 日 |
| 增加了對加拿大 (中部) (CA-中央-1) 的額外可用區域的支持。 | Amazon DocumentDB 現在可在加拿大 (中部) 區域 (CA-中央-1) 使用 R5 類別執行個體和 3 個可用區域使用。 | 2020 年 3 月 26 日 |

| | | |
|--|--|------------------|
| 增加了對兩個額外的 MongoDB API 的支持。 | Amazon DocumentDB 增加了對\$dateFromString 和 executionStats MongoDB 的 API 的支持。 | 2020 年 3 月 23 日 |
| 增加了對五個額外的 MongoDB API 的支持。 | Amazon DocumentDB 添加了對\$objectToArray、\$arrayToObject、\$slice、\$mod和\$range MongoDB API 的支持。 | 2020 年 2 月 6 日 |
| 增加了對加拿大 (中部) 的支持。 | Amazon DocumentDB 現在可在加拿大 (中部) 區域 (CA-中央 -1) 搭配 R5 類別執行個體使用。 | 2019 年 12 月 11 日 |
| 增加了對 ChangeStreamLogSize。 | Amazon DocumentDB 添加了對雲觀察指標的ChangeStreamLogSize 支持。 | 2019 年 11 月 22 日 |
| 增加了對歐洲 (巴黎) 地區的支持 | Amazon DocumentDB 現在可在歐洲 (巴黎) 區域 (歐洲至西部 -3) 使用 R5 類別執行個體使用。 | 2019 年 10 月 30 日 |
| 增加了對亞太區域 (孟買) 地區的支持 | Amazon DocumentDB 現在可在亞太區域 (孟買) 區域 (距離南 -1) 搭配 R5 類別執行個體使用。 | 2019 年 10 月 17 日 |
| 增加了對三個額外的 MongoDB API 的支持 | Amazon DocumentDB 增加了對\$addFields、\$concatArrays 和 \$lookup MongoDB API 的支持。 | 2019 年 10 月 16 日 |

| | | |
|--------------------------------------|---|------------------|
| 增加了對亞太區域 (新加坡) 區域的支持 | Amazon DocumentDB 現在可在亞太區域 (新加坡) 區域 (AP-東南 -1) 搭配 R5 類別執行個體使用。 | 2019 年 10 月 14 日 |
| 添加了更新 TLS 證書的新文檔 | 新增更新 CA 憑證以使用新的 CA 憑證建立 TLS 連線的說明。 | 2019 年 10 月 2 日 |
| 增加了對證書的 API 支持 | Amazon DocumentDB 適用於執行個體的新憑證資料類型。如需詳細資訊，請參閱 DBInstance 。 | 2019 年 10 月 1 日 |
| Support 查詢分析 | Amazon DocumentDB 新增了對叢集執行個體和資料庫上支援的操作進行分析的功能。 | 2019 年 8 月 19 日 |
| 在亞太地區 (東京) 增加了第三個 AZ | 亞 Amazon DocumentDB 為您在亞太區域 (東京) 的運算執行個體新增了第三個可用區域 (AZ)。 | 2019 年 8 月 9 日 |
| Support 其他蒙戈 API | 已新增對其他彙總管線功能的支援 <code>\$in</code> ，包括 <code>\$isoWeek</code> <code>\$isoWeekYear</code> 、 <code>\$isoDayOfWeek</code> 、和 <code>\$dateToString</code> 彙總運算子以及 <code>\$addToSet</code> 彙總階段。Amazon DocumentDB 也新增了對集合層級診斷 <code>top()</code> 命令的支援，以及使用該命令修改 TTL 索引 <code>expireAfterSeconds</code> 參數的功能。 <code>collMod()</code> | 2019 年 7 月 31 日 |

| | | |
|-----------------------------------|---|-----------------|
| 增加了對歐洲 (倫敦) 的支持 | Amazon DocumentDB 現在可在歐洲 (倫敦) (歐洲至西部 -2) 使用 R5 類別執行個體使用。 | 2019 年 7 月 18 日 |
| 添加的代碼示例 | 在 R 和 Ruby 中新增程式碼範例，以程式設計方式連接至 Amazon DocumentDB。 | 二零一九年七月十七 |
| 新增最佳做法 | 新增了最佳實務，可協助您管理 Amazon DocumentDB 成本。 | 二零一九年七月十七 |
| Support 停止和啟動叢集 | Amazon DocumentDB 新增了停止和啟動叢集的支援，以管理開發和測試環境的成本。 | 2019 年 7 月 1 日 |
| Support 叢集刪除保護 | 為了保護您的叢集免於意外刪除，Amazon DocumentDB 新增了刪除保護功能。 如需詳細資訊，請參閱下列主題：建立 Amazon DocumentDB 叢集、修改 Amazon DocumentDB 叢集、刪除 Amazon DocumentDB 叢集，以及 DeletionProtection API 主題資料庫叢集。 | 2019 年 7 月 1 日 |
| 功能差異更新 | 在功能差異中新增了隱含交易。 | 2019 年 6 月 26 日 |
| 功能差異添加 | 添加了有關 Amazon DocumentDB 中存儲和索引壓縮的注意事項。 | 2019 年 6 月 13 日 |
| 支援其他地區 | 亞 Amazon DocumentDB 現在可在亞太區域 (雪梨) (AP-東南 -2) 使用 R5 類別執行個體使用。 | 2019 年 6 月 5 日 |

| | | |
|----------------------------------|---|-----------------|
| 其他區域支援 R5 執行個體類別 | 已新增 4 個其他區域支援 R5 執行個體類別：美國東部 (俄亥俄州)、美國東部 (維吉尼亞北部)、美國西部 (奧勒岡) 和歐洲 (愛爾蘭)。透過這項變更，所有支援 Amazon DocumentDB 的區域都支援 R5 執行個體。 | 2019 年 5 月 17 日 |
| 支援其他地區 | 增加了對 2 個其他區域的支援，包括具有 R5 執行個體類別的亞太區域 (東京) (ap-Northeast-1) 和亞太區域 (首爾) (AP-Northeast-2)。如需詳細資訊，請參閱 各區域支援的執行個體類別 和 執行個體類別規格 。 | 2019 年 5 月 8 日 |
| 增加了更多連接代碼示例 | 在 Java 和 C# 中添加了用於連接到 Amazon DocumentDB 的代碼示例。 | 2019 年 4 月 24 日 |
| 額外的蒙戈 API 支持 | 已新增下列支援：7 個彙總字串運算子 (\$indexOfBytes、\$indexOfCp、\$strlenBytes、\$strlenCp、\$toLowerCase、\$toUpperCase 和 \$split)、9 個日期時間運算子 (\$dayOfYear、\$dayOfMonth、\$dayOfWeek、\$year、\$month、\$hour、\$minute、\$second 和 \$millisecond)，以及 \$sample 彙總管道階段。 | 2019 年 4 月 4 日 |
| 新增連線程式碼範例 | 在 Python，Node.js，PHP 和 Go 中添加了代碼示例，用於連接到 Amazon DocumentDB。 | 2019 年 3 月 21 日 |

| | | |
|---|---|-----------------|
| Support 法蘭克福區域和 R5 執行個體 | 使用 R5 執行個體類別新增了對歐洲 (法蘭克福) 區域 (歐盟中部 -1) 的支援。如需詳細資訊，請參閱 各區域支援的執行個體類別 和 執行個體類別規格 。 | 2019 年 3 月 13 日 |
| 匯總管道運算符支持 | 已新增下列支援：新的彙總字串運算子 (\$concat、\$substr、\$substrBytes、\$substrCP、\$strcasecmp)、陣列彙總運算子 (\$size)、彙總群組累加運算子 (\$push) 和彙總階段 (\$redact 和 \$indexStats)。我們也新增了位置陣列運算子 (\$[] 和 \$[<identifier>]) 及 hint() 的支援。 | 2019 年 2 月 28 日 |
| 引擎升級 | 已新增文件，以判斷待定叢集修改和升級您叢集的引擎版本。 | 2019 年 2 月 15 日 |
| 稽核事件 | 增加了對使用 Amazon CloudWatch 日誌稽核資料庫事件的支援。 | 2019 年 2 月 12 日 |
| Quick Start | 添加了一個快速入門主題，以幫助您輕鬆地開始使用 Amazon DocumentDB。AWS CloudFormation | 2019 年 1 月 11 日 |

[公開發行](#)

這是 Amazon DocumentDB B 的初始公開發行版本 (與 MongoDB 兼容性)。此版本包括 [開發人員指南](#) 和整合的 [資源管理 API 參考](#)。

2019 年 1 月 9 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。