



PostgreSQL

Enterprise applications

NetApp
January 02, 2026

This PDF was generated from <https://docs.netapp.com/zh-tw/ontap-apps-dbs/postgres/postgres-overview.html> on January 02, 2026. Always check docs.netapp.com for the latest.

目錄

PostgreSQL	1
總覽	1
資料庫組態	1
架構	1
初始化參數	2
設定	3
表格空間	4
儲存組態	4
NFS	4
SAN	6
資料保護	7
原生 Data 保護	7
快照	8
資料保護軟體	8

PostgreSQL

總覽

PostgreSQL 隨附的變種包括 PostgreSQL 、 PostgreSQL Plus 和 EDBS PostgreSQL 進階伺服器 (EPAS) 。 PostgreSQL 通常部署為多層應用程式的後端資料庫。它受一般中介軟體套件的支援 (例如 PHP 、 Java 、 Python 、 Tcl/Tk 、 ODBC 、和 JDBC) ，過去一直是開放原始碼資料庫管理系統的熱門選擇。ONTAP 是執行 PostgreSQL 資料庫的絕佳選擇、可確保其可靠性、高效能及高效率的資料管理功能。



ONTAP 和 PostgreSQL 資料庫上的這份文件取代先前發佈的 _TR-4770 : ONTAP 最佳實務做法的 PostgreSQL 資料庫。 _

隨著資料呈指數成長、企業的資料管理變得更加複雜。這種複雜性會增加授權、營運、支援和維護成本。若要降低整體 TCO 、請考慮使用可靠、高效能的後端儲存設備、從商業資料庫切換為開放原始碼資料庫。

ONTAP 是理想的平台、因為 ONTAP 是專為資料庫所設計。為了滿足資料庫工作負載的需求、我們特別建立了許多功能、例如隨機 IO 延遲最佳化、以提供進階服務品質 (QoS) 到基本 FlexClone 功能。

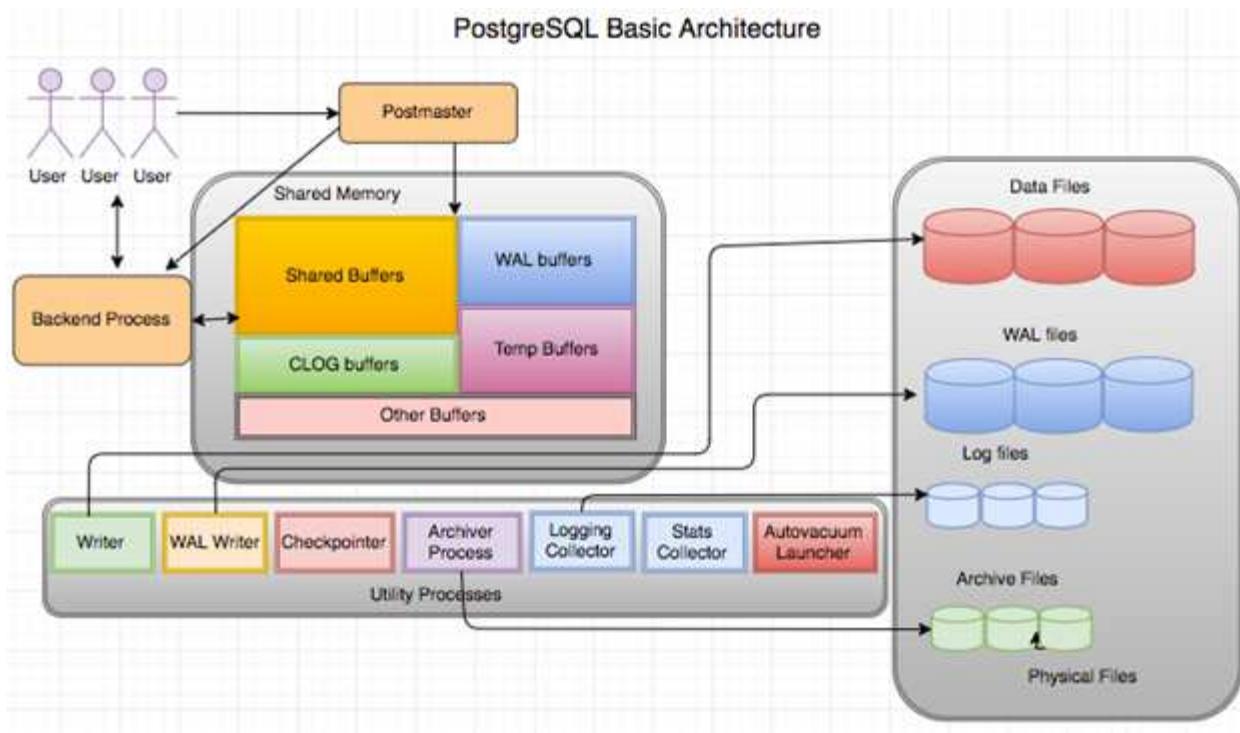
其他功能 (例如不中斷升級) (包括儲存設備更換) 、可確保關鍵資料庫仍可使用。您也可以透過 MetroCluster 為大型環境進行即時災難恢復、或是使用 SnapMirror 主動式同步來選擇資料庫。

最重要的是、ONTAP 提供無與倫比的效能、並能根據您的獨特需求調整解決方案的規模。我們的高階系統可提供超過 1M IOPS 、延遲時間以微秒為測量單位、但如果只需要 10 萬次 IOPS 、您可以使用仍執行相同儲存作業系統的較小控制器來調整儲存解決方案的大小。

資料庫組態

架構

PostgreSQL 是以用戶端和伺服器架構為基礎的 RDBMS 。 PostgreSQL 執行個體稱為資料庫叢集、是資料庫的集合、而非伺服器集合。



PostgreSQL 資料庫中有三個主要元素：郵政局長、前端（用戶端）和後端用戶端會傳送要求給郵政局長、其中包含 IP 傳輸協定等資訊、以及要連線的資料庫。郵政局長會驗證連線、並將其傳送至後端程序以進行進一步的通訊。後端程序會執行查詢、並將結果直接傳送至前端（用戶端）。

PostgreSQL 執行個體是以多重處理模式為基礎、而非多重執行緒模式。它會為不同的工作產生多個處理程序、而且每個處理程序都有自己的功能。主要程序包括用戶端程序、Wal 寫入程序、背景寫入程序及檢查指標程序：

- 當用戶端（前景）程序傳送讀取或寫入要求至 PostgreSQL 執行個體時、它不會直接讀取或寫入資料至磁碟。它首先緩衝共享緩衝區和預先寫入記錄（Wal）緩衝區中的資料。
- Wal 寫入器程序會操控共用緩衝區和 Wal 緩衝區的內容、以寫入 Wal 記錄檔。Wal 記錄檔通常是 PostgreSQL 的交易記錄檔、並依序寫入。因此、為了改善資料庫的回應時間、PostgreSQL 會先寫入交易記錄檔、並確認用戶端。
- 若要將資料庫置於一致的狀態、背景寫入器程序會定期檢查共用緩衝區是否有髒頁。然後將資料排清至儲存在 NetApp 磁碟區或 LUN 上的資料檔案。
- checkpoint 程序也會定期執行（比背景程序更少）、並防止對緩衝區進行任何修改。它會向 Wal 寫入器程序發出訊號、將檢查點記錄寫入並清除至儲存在 NetApp 磁碟上的 Wal 記錄檔結尾。它也會向背景寫入器程序發出訊號、要求將所有髒頁寫入磁碟並清除。

初始化參數

您可以使用建立新的資料庫叢集 `initdb` 方案。一 `initdb` 指令碼會建立定義叢集的資料檔案、系統表格和範本資料庫（`template0` 和 `template1`）。

範本資料庫代表常用資料庫。其中包含系統表格、標準檢視、功能和資料類型的定義。`pgdata` 做為的引數 `initdb` 指定資料庫叢集位置的指令碼。

PostgreSQL 中的所有資料庫物件都是由各自的 OID 在內部管理。表格和索引也由個別的 OID 管理。資料庫物件與其各自的 OID 之間的關係會儲存在適當的系統目錄表格中、視物件類型而定。例如、資料庫和堆積表格的

OID 會儲存在中 `pg_database` 和 "`pg_class`" 。您可以在 PostgreSQL 用戶端上發佈查詢來判斷 OID 。

每個資料庫都有自己的個別資料表和索引檔案、限制為 1GB 。每個表格都有兩個相關的檔案、分別以後綴表示 `_fsm` 和 `_vm`。它們稱為可用空間地圖和可見度地圖。這些檔案會儲存可用空間容量的相關資訊、並在表格檔案中的每個頁面上都有可見度。索引只有個別的可用空間地圖、而且沒有可見度地圖。

◦ `pg_xlog/pg_wal` 目錄包含預先寫入記錄。預先寫入記錄可用來改善資料庫的可靠性和效能。每當您更新表格中的列時、PostgreSQL 會先將變更寫入預先寫入記錄、然後將修改寫入實際的資料頁面到磁碟。。

`pg_xlog` 目錄通常包含數個檔案、但 `initdb` 只會建立第一個檔案。視需要新增額外檔案。每個 `xlog` 檔案長度為 16MB 。

設定

有幾種 PostgreSQL 調校組態可以改善效能。

最常用的參數如下：

- `max_connections = <num>`：一次擁有的最大資料庫連線數。使用此參數可限制磁碟交換並終止效能。根據應用程式需求、您也可以針對連線集區設定調整此參數。
- `shared_buffers = <num>`：提高資料庫伺服器效能的最簡單方法。對於大多數現代硬體而言、預設值為低。在部署期間、系統上的可用 RAM 約為 25%。此參數設定會因其與特定資料庫執行個體的運作方式而異；您可能必須根據試用和錯誤來增加和減少值。不過、將其設為高可能會降低效能。
- `effective_cache_size = <num>`：此值告訴 PostgreSQL 的最佳化程式 PostgreSQL 有多少記憶體可用於快取資料、並有助於判斷是否使用索引。較大的值會增加使用索引的可能性。此參數應設為分配給的記憶體容量 `shared_buffers` 加上可用的作業系統快取容量。此值通常超過系統總記憶體的 50% 。
- `work_mem = <num>`：此參數控制用於排序作業和雜湊表的記憶體容量。如果您在應用程式中進行大量排序、可能需要增加記憶體容量、但請謹慎。它不是系統範圍的參數、而是每次操作的參數。如果複雜查詢中有多個排序作業、則會使用多個 `work_mem` 記憶體單元、而多個後端也可能同時執行此作業。如果值太大、此查詢通常會引導您的資料庫伺服器進行切換。此選項先前在舊版 PostgreSQL 中稱為 `sort_mem` 。
- `fsync = <boolean> (on or off)`：此參數確定在提交事務之前是否應使用 `fsync ()` 將所有 Wal 頁面同步到磁盤。關閉它有時會改善寫入效能、並將其開啟、以提高系統當機時避免毀損的風險。
- `checkpoint_timeout`：檢查點處理程序會將已提交的資料清除至磁碟。這涉及磁碟上的大量讀寫作業。此值以秒為單位設定、較低的值可減少損毀恢復時間、而增加的值則可減少檢查點呼叫、進而降低系統資源的負載。根據應用程式的關鍵程度、使用量、資料庫可用度、設定 `checkpoint` 過時的值。
- `commit_delay = <num>` 和 `commit_siblings = <num>`：這些選項可同時用於撰寫多筆同時提交的交易、以協助改善效能。如果交易提交時有多個 `command_siblings` 物件處於作用中狀態、伺服器會等待 `command_delay` 微秒、嘗試一次提交多個交易。
- `max_worker_processes / max_parallel_workers`：配置流程的最佳工作人員數量。`max_parallel_workers` 對應可用的 CPU 數量。視應用程式設計而定、查詢可能需要較少的工作人員來執行平行作業。最好保持兩個參數的值相同、但在測試後調整值。
- `random_page_cost = <num>`：此值控制 PostgreSQL 檢視非連續磁碟讀取的方式。較高的值表示 PostgreSQL 較可能使用連續掃描、而非索引掃描、表示伺服器有快速磁碟。請在評估其他選項（例如計畫型最佳化、吸塵、索引以變更查詢或架構）之後、修改此設定。
- `effective_io_concurrency = <num>`：此參數設置 PostgreSQL 嘗試同時執行的並行磁盤 I/O 操作數。提高此值會增加任何個別 PostgreSQL 工作階段嘗試平行啟動的 I/O 作業數。允許範圍為 1 到 1,000、或為零、以停用非同步 I/O 要求的發出。目前、此設定只會影響點陣圖堆疊掃描。固態硬碟（SSD）和其他記憶體型儲存設備（NVMe）通常可以處理許多並行要求、因此最佳價值可能在數百種環境中。

如需 PostgreSQL 組態參數的完整清單、請參閱 PostgreSQL 文件。

吐司

Toast 代表「超大型屬性儲存技術」。PostgreSQL 使用固定的頁面大小（通常為 8KB）、「不允許 Tuple 跨越多個頁面」。因此、無法直接儲存大欄位值。當您嘗試儲存超過此大小的資料列時、Toast 會將大型資料欄的資料分成較小的「片段」、並將其儲存在 Toast 資料表中。

只有在將結果集傳送至用戶端時、才會拔出（如果完全選取）已烘烤的屬性大值。表格本身比沒有任何離線儲存設備（Toast）時更小、可容納更多資料列到共用緩衝區快取中。

真空

在正常的 PostgreSQL 作業中、因為更新而刪除或過時的 Tuple 不會從其表格中實際移除、直到執行真空為止。因此、您必須定期執行吸氣、尤其是在經常更新的表格上。接著必須回收它所佔用的空間、以便新列重複使用、以避免磁碟空間中斷。不過、它不會將空間傳回作業系統。

頁面內的可用空間不會分散。真空會重新寫入整個區塊、有效地將剩餘的資料列封裝起來、並在頁面中留下一個連續的可用空間區塊。

相反地、使用「完全真空」技術、可以撰寫完全新版的表格檔案、而不會有任何空間。此動作可將表格大小減至最小、但可能需要很長時間。在作業完成之前、它也需要額外的磁碟空間來容納表格的新複本。例行真空的目標是避免完全真空。此程序不僅能將資料表保持在最小大小、還能維持磁碟空間的穩定狀態使用量。

表格空間

初始化資料庫叢集時會自動建立兩個資料表空間。

◦ pg_global 表空間用於共享系統目錄。◦ pg_default 表空間是 template1 和 template0 資料庫的預設資料表空間。如果初始化叢集的磁碟分割或磁碟區空間不足且無法擴充、則可在不同的磁碟分割上建立資料表空間、並在重新設定系統之前使用。

大量使用的索引可以放在快速、高可用度的磁碟上、例如固態裝置。此外、儲存歸檔資料的表格、無論是極少使用或非效能關鍵、都可以儲存在成本較低、速度較慢的磁碟系統上、例如 SAS 或 SATA 磁碟機。

資料表空間是資料庫叢集的一部分、無法視為資料檔案的獨立集合。它們取決於主資料目錄中包含的中繼資料、因此無法附加至不同的資料庫叢集或個別備份。同樣地、如果您遺失資料表空間（透過檔案刪除、磁碟故障等方式）、資料庫叢集可能會變得無法讀取或無法啟動。將資料表空間放在像 RAM 磁碟一樣的暫存檔案系統上、會危及整個叢集的可靠性。

建立表格後、如果要求的使用者擁有足夠的權限、則可以從任何資料庫使用表格區。PostgreSQL 使用符號連結來簡化資料表空間的實作。PostgreSQL 會在中新增一列 pg_tablespace 表（叢集範圍表格）、並將新的物件識別碼（OID）指派給該列。最後、伺服器會使用 OID 在叢集與指定目錄之間建立符號連結。目錄 \$PGDATA/pg_tblspc 包含指向叢集中定義的每個非內建表格空間的符號連結。

儲存組態

NFS

PostgreSQL 資料庫可以裝載在 NFSv3 或 NFSv4 檔案系統上。最佳選項取決於資料庫以外的因素。

例如、在某些叢集式環境中、NFSv4 鎖定行為可能較為理想。（請參閱 ["請按這裡"](#) 如需其他詳細資料）

否則、資料庫功能應接近相同、包括效能。唯一的要求是使用 `hard` 掛載選項。這是為了確保軟性逾時不會產生無法恢復的 IO 錯誤。

如果選擇 NFSv4 作為傳輸協定、NetApp 建議使用 NFSv4.1。NFSv4.1 中的 NFSv4 傳輸協定有一些功能性增強功能、可改善 NFSv4.0 的恢復能力。

針對一般資料庫工作負載、請使用下列掛載選項：

```
rw,hard,nointr,vers=[3|4],proto=tcp,rsize=65536,wsize=65536
```

如果需要大量連續 IO、則可依照下節所述增加 NFS 傳輸大小。

NFS 傳輸大小

根據預設、ONTAP 將 NFS I/O 大小限制為 64K。

大多數應用程式和資料庫的隨機 I/O 使用的區塊大小要小得多、遠低於 64K 的最大值。大型區塊 I/O 通常是平行處理的、因此 64K 的最大值也不是取得最大頻寬的限制。

有些工作負載的上限為 64K、因此會造成限制。特別是、如果資料庫執行的 I/O 數量較少、但容量較大、則備份或還原作業或資料庫完整表格掃描等單執行緒作業、會更快、更有效率地執行。ONTAP 的最佳 I/O 處理大小為 256K。

指定 ONTAP SVM 的最大傳輸大小可變更如下：

```
Cluster01::> set advanced
Warning: These advanced commands are potentially dangerous; use them only
when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y
Cluster01::*> nfs server modify -vserver vserver1 -tcp-max-xfer-size
262144
Cluster01::*>
```

切勿將 ONTAP 上允許的傳輸大小上限降至低於目前掛載之 NFS 檔案系統的 `rsize`/`wsize` 值。這可能會在某些作業系統中造成當機或甚至資料毀損。例如、如果 NFS 用戶端目前設定為 `rsize`/`wsize` 65536、則 ONTAP 最大傳輸大小可在 65536 到 1048576 之間調整、因為用戶端本身受到限制、因此沒有任何影響。將傳輸大小上限降至 65536 以下可能會損害可用度或資料。

在 ONTAP 層級增加傳輸大小後、將會使用下列掛載選項：

```
rw,hard,nointr,vers=[3|4],proto=tcp,rsize=262144,wsize=262144
```

NFSv3 TCP 插槽表

如果 NFSv3 與 Linux 搭配使用、則正確設定 TCP 插槽表至關重要。

TCP 插槽表是與主機匯流排介面卡（HBA）併列深度相當的 NFSv3。這些表格可控制任何時間都可以處理的 NFS 作業數量。預設值通常為 16、這對於最佳效能而言太低。相反的問題發生在較新的 Linux 核心上、這會自動將 TCP 插槽表格限制增加到要求使 NFS 伺服器飽和的層級。

為了達到最佳效能並避免效能問題、請調整控制 TCP 插槽表的核心參數。

執行 `sysctl -a | grep tcp.*.slot_table` 並觀察下列參數：

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

所有 Linux 系統都應該包括在內 `sunrpc.tcp_slot_table_entries`、但只有部分包含在內 `sunrpc.tcp_max_slot_table_entries`。兩者都應設為 128。

 若未設定這些參數、可能會對效能造成重大影響。在某些情況下、效能會受到限制、因為 Linux 作業系統沒有發出足夠的 I/O 在其他情況下、隨著 Linux 作業系統嘗試發出的 I/O 數量超過可服務的數量、I/O 延遲也會增加。

SAN

具有 SAN 的 PostgreSQL 資料庫通常裝載在 xfs 檔案系統上、但如果作業系統廠商支援、也可以使用其他資料庫

雖然單一 LUN 通常可支援高達十萬 IOPS、但 IO 密集的資料庫通常需要使用含分段的 LVM。

LVM 分拆

在快閃磁碟機時代之前、使用區塊延展來協助克服旋轉磁碟機的效能限制。例如、如果作業系統需要執行 1MB 讀取作業、則從單一磁碟機讀取 1MB 的資料時、需要大量的磁碟機磁頭搜尋和讀取、因為 1MB 會緩慢傳輸。如果將 1MB 的資料分散在 8 個 LUN 上、則作業系統可能會同時執行 8 個 128K 讀取作業、並縮短完成 1MB 傳輸所需的時間。

由於必須事先知道 I/O 模式、因此使用旋轉磁碟機進行分拆會更困難。如果串列區塊延展未針對真正的 I/O 模式正確調整、則等量區塊配置可能會損害效能。使用 Oracle 資料庫、特別是搭配 All Flash 組態、分拆作業更容易設定、並經證實可大幅提升效能。

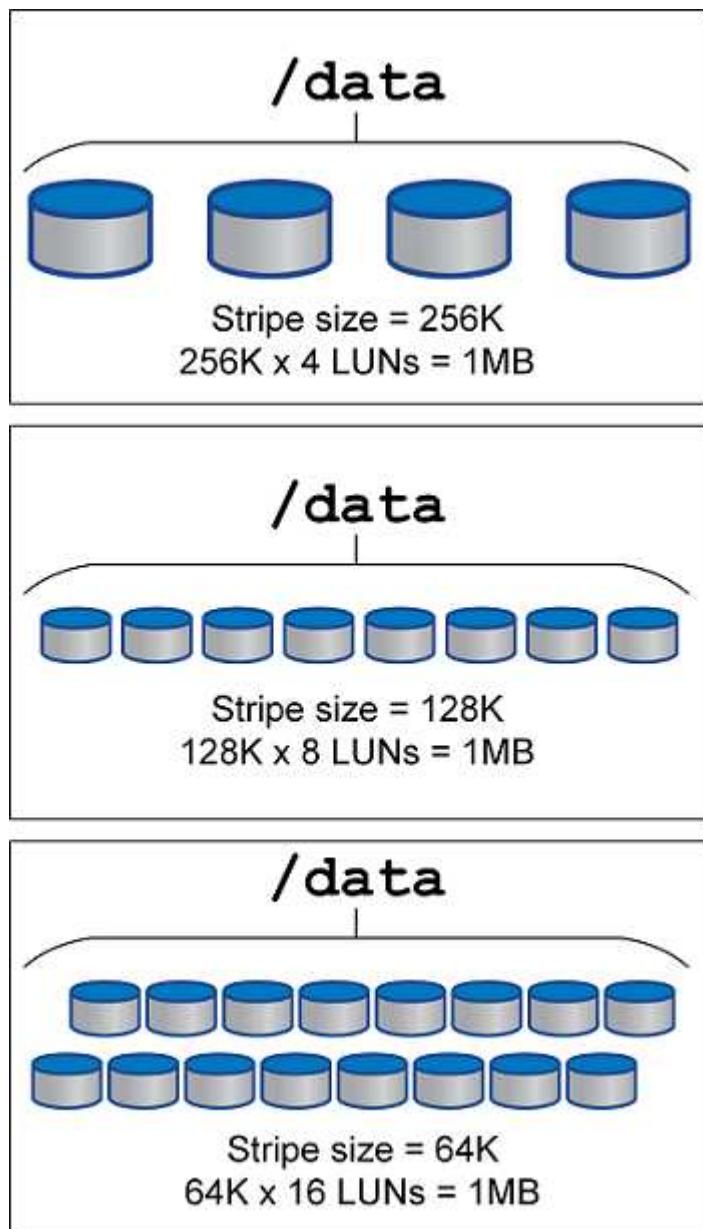
依預設、邏輯磁碟區管理程式（例如 Oracle ASM 等量磁碟區）不屬於原生 OS LVM。其中有些 LUN 會將多個 LUN 連結在一起、成為串連的裝置、導致資料檔案存在於一台 LUN 裝置上、而只存在於一台 LUN 裝置上。這會造成熱點。其他 LVM 實作預設為分散式擴充。這與分拆類似、但卻是比較粗糙的。磁碟區群組中的 LUN 會切成大型片段、稱為區段、通常以百萬位元組為單位測量、然後邏輯磁碟區會分佈在這些區段中。結果是對檔案進行隨機 I/O、應該能在 LUN 之間妥善分配、但連續 I/O 作業的效率卻不如以前那麼高。

效能密集的應用程式 I/O 幾乎總是（a）以基本區塊大小為單位、或（b）1 MB。

等量分配組態的主要目標是確保單一檔案 I/O 可作為單一單元執行、而多區塊 I/O 的大小應為 1MB、可在等量

磁碟區中的所有 LUN 之間平均平行處理。這表示等量磁碟區大小不得小於資料庫區塊大小、且等量磁碟區大小乘以 LUN 數量應為 1MB 。

下圖顯示等量磁碟區大小和寬度調校的三個可能選項。選擇 LUN 數量以滿足上述效能需求、但在所有情況下、單一等量磁碟區內的總資料為 1MB 。



資料保護

原生 Data 保護

儲存設計的主要層面之一、是為 PostgreSQL Volume 提供保護。客戶可以使用傾印方法或使用檔案系統備份來保護 PostgreSQL 資料庫。本節說明備份個別資料庫或整個叢集的不同方法。

備份 PostgreSQL 資料有三種方法：

- SQL Server 備份
- 檔案系統層級備份
- 持續歸檔

SQL Server 備份方法背後的想法是使用 SQL Server 命令產生檔案、當檔案傳回伺服器時、就能像在備份時一樣重新建立資料庫。PostgreSQL 提供公用程式 `pg_dump` 和 `pg_dump_all` 用於建立個別和叢集層級的備份。這些備份都是邏輯的、而且沒有足夠的資訊可供 Wal 重新執行使用。

另一種備份策略是使用檔案系統層級的備份、系統管理員可在其中直接複製 PostgreSQL 用來將資料儲存在資料庫中的檔案。此方法是在離線模式下執行：必須關閉資料庫或叢集。另一種方法是使用 `pg_basebackup` 執行 PostgreSQL 資料庫的熱串流備份。

快照

PostgreSQL 的 Snapshot 型備份需要為資料檔案、Wal 檔案和歸檔的 Wal 檔案組態快照、才能提供完整或時間點還原。

對於 PostgreSQL 資料庫而言、快照的平均備份時間介於數秒到數分鐘之間。此備份速度比快 60 至 100 倍 `pg_basebackup` 以及其他檔案系統備份方法。

NetApp 儲存設備上的快照既可一致當機、也可一致應用程式。當機一致的快照是在儲存設備上建立、而不會停止資料庫、而在資料庫處於備份模式時則會建立應用程式一致的快照。NetApp 也可確保後續快照是永久遞增備份、以節省儲存成本並提高網路效率。

由於快照速度很快、不會影響系統效能、因此您可以每天排程多個快照、而不必像其他串流備份技術一樣建立單一的每日備份。當需要還原與還原作業時、系統停機時間會減少兩項重要功能：

- NetApp SnapRestore 資料還原技術代表還原作業只需數秒即可完成。
- 積極的恢復點目標（RPO）意味著必須套用較少的資料庫記錄、同時也加速轉送恢復。

若要備份 PostgreSQL、您必須使用（一致性群組）Wal 和歸檔記錄、確保資料磁碟區同時受到保護。使用 Snapshot 技術複製 Wal 檔案時、請務必執行 `pg_stop` 清除所有必須歸檔的 Wal 項目。如果您在還原期間清除 Wal 項目、則只需要停止資料庫、卸載或刪除現有的資料目錄、並在儲存設備上執行 SnapRestore 作業。還原完成後、您可以掛載系統並將其恢復至目前狀態。對於時間點恢復、您也可以還原 Wal 和歸檔記錄檔、然後 PostgreSQL 會決定最一致的點並自動恢復。

一致性群組是 ONTAP 中的一項功能、建議在有多個磁碟區掛載到單一執行個體或具有多個資料表空間的資料庫時使用。一致性群組快照可確保將所有磁碟區分組並加以保護。一致性群組可從 ONTAP 系統管理員有效管理、您甚至可以複製它來建立資料庫的執行個體複本、以供測試或開發之用。

資料保護軟體

適用於 PostgreSQL 資料庫的 NetApp SnapCenter 外掛程式、結合 Snapshot 和 NetApp FlexClone 技術、可為您提供下列優點：

- 快速備份與還原。
- 節省空間的複本。
- 能夠建置快速有效的災難恢復系統。

在下列情況下、您可能偏好選擇 NetApp 的優質備份合作夥伴、例如 Veeam Software 和 CommVault：

- 在異質環境中管理工作負載
- 將備份儲存至雲端或磁帶、以供長期保留
- 支援多種作業系統版本和類型

適用於 PostgreSQL 的 SnapCenter 外掛程式是社群支援的外掛程式、可在 NetApp Automation Store 取得設定與文件。透過 SnapCenter、使用者可以遠端備份資料庫、複製及還原資料。

版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP 「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。