AWS白皮书

AWS 多區域基礎知識

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 多區域基礎知識: AWS白皮书

Copyright © 2023 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，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，或由 Amazon 贊助。

Table of Contents

AWS 多區域基礎知識

出版日期：二零二二年十二月二十日（文件修訂）

摘要

這份進階的 300 層級 paper 適用於建置工作負載的雲端架構師和資深領導者，他們有興趣使用多區域架構來提高工作負載的彈性。AWS本 paper 假設基礎設AWS施和服務的基本知識。它概述了常見的多區域使用案例，分享多區域的基本概念和設計、開發和部署方面的影響，並提供規範性指導，協助您更好地判斷多區域架構是否適合您的工作負載。

你是否 Well-Architected？

AWSWell-Architected 的架構可協助您瞭解在雲端中建置系統時所做決策的優缺點。Framework 的六大支柱可讓您學習如何設計和操作可靠、安全、高效、符合成本效益且可持續發展的系統的架構最佳實務。使用中免費提供的 AWS Well-Architected ToolAWS Management Console，您可以針對每個支柱回答一組問題，根據這些最佳實務來檢閱工作負載。

如需雲端架構的更多專家指導和最佳實務 (參考架構部署、圖表和白皮書)，請參閱架構中心。AWS

簡介

每個區域AWS 區域由地理區域內的多個獨立且實際上獨立的可用區域組成。維護每個區域中的軟件服務之間的嚴格邏輯分離。這種有目的的的設計可確保一個區域中的基礎設施或服務故障不會導致另一個區域的相關故障。

大多數AWS客戶可以使用多個可用區域 (AZ) 或區域AWS服務，在單一區域中達成工作負載的彈性目標。不過，部分客戶採用多區域架構的原因有三個。

- 對於他們認為在單一區域中無法滿足的最高層級工作負載，這些工作負載具有高可用性和持續性的營運需求。
- 他們需要滿足數據主權要求（例如遵守當地法律，法規和合規性），這些要求工作負載才能在特定司法管轄區內運行。
- 他們需要在最接近最終使用者的位置執行工作負載，以改善工作負載的效能和客戶體驗。

本 paper 著重於高可用性和營運需求的持續性，並協助您瞭解針對工作負載採用多區域架構的考量。我們將說明適用於設計、開發和部署多區域工作負載的基本概念，以及可協助您判斷多區域架構是否適

合特定工作負載的規範性架構。您需要確保多區域架構是工作負載的正確選擇，因為這些架構具有挑戰性，而且如果未正確完成，工作負載的整體可用性可能會降低。

在單一區域進行工程設計和作業的彈性

在深入探討多區域概念之前，請先確認您的工作負載已在單一區域中盡可能具備彈性。若要達成此目標，請根據 AWS Well-Architected 架構的可靠性支柱和卓越營運支柱評估您的工作負載，並進行任何必要的變更，以採用建議的最佳實務。AWSWell-Architected 的框架涵蓋了以下概念：

- 根據網域界限分割工作負載
- 明確定義的服務合約
- 依賴管理和耦合
- 處理失敗、重試和退出策略
- 冪等操作和有狀態與無狀態交易
- 營運準備和變更管理
- 瞭解工作負載健
- 回應事件

若要進一步採用單一區域彈性，請檢閱並套用進階異地同步備份復原模式中討論的概念，以處理灰色故障。本 paper 提供有關在每個可用區域中使用複本以遏止故障的最佳實務，並擴展 W AWS ell Architected 中引入的異地同步備份概念。一旦您完全套用建議的概念和最佳實務，以達到單一區域中的最高彈性，就可以根據多區域架構的基本原理來評估特定工作負載，以判斷是否可以使用多區域方法提高工作負載的彈性。

多區域基礎 1：了解需求

如前所述，高可用性和營運持續性是採用多區域架構的常見原因。可用性指標會測量工作負載在定義期間內可供使用的時間百分比，而作業連續性指標則衡量大規模和通常持續時間較長事件的復原。

測量可用性幾乎是一個連續的過程。特定的測量或指標可能會有所不同，但通常會圍繞目標可用性結合，通常稱為 Nines (例如 99.99% 可用性)。根據可用性目標，一種尺寸並不適合所有人。可用性目標需要在工作負載層級建立，而不是在所有工作負載中套用單一目標，將非關鍵元件與重要元件分開。

對於操作的連續性，通常使用以下 point-in-time測量：

- 復原時間目標 (RTO) — RTO 是服務中斷與恢復服務之間可接受的最大延遲。此值決定服務受損的可接受持續時間。
- 復原點目標 (RPO) — RPO 是自上次資料復原點以來可接受的時間上限。這會決定最新復原點與服務中斷之間，哪些資料會被視為可接受的資料遺失。

與設定可用性目標類似，RTO 和 RPO 也應在工作負載層級定義。為了實現更積極的營運連續性或高可用性要求，需要增加投資。也就是說，並非每個應用程序都可以要求或需要相同級別的彈性。建立分層機制可協助建立架構，讓企業和 IT 擁有者能夠根據業務影響識別需求最高的應用程式，並據此分層。分層的範例可在下表中找到。

表 1 — SLA 的彈性分層範例

可用性服務等級協定 (SLA)	彈性層級	可接受的停機時間/年
99.99%	鉑	二百六十分鐘
99.90%	黃金	晚上七十七小時
99.5%	銀色	八十三天

表 2 — RTO 和 RPO 的彈性分層範例

層	最大 RTO	最大投資回收	條件	費用
鉑	15 分鐘	五分鐘	任務關鍵性工作	$$$

層	最大 RTO	最大投資回收	條件	費用
黃金	十五分鐘至六小時	兩小時	重要但非關鍵任務工作負載	$$
銀色	六小時 — 幾天	24 小時	非關鍵工作負載	$

在設計工作負載以提供彈性時，必須瞭解高可用性與作業持續性之間的關係。例如，如果工作負載需要 99.99% 的可用性，則每年不可容忍超過 53 分鐘的停機時間。偵測故障可能需要至少五分鐘，操作員可能需要 10 分鐘的時間才能參與、做出復原步驟的決策，以及執行這些步驟。單個問題需要 30 到 45 分鐘才能恢復並不罕見。在這種情況下，使用多區域策略提供隔離的執行個體以消除相關影響，可以通過在有限的時間內容錯移轉來繼續操作，同時獨立分類初始減值。這是必要定義適當的 RTO 和 RPO 的地方。

對於具有極高可用性需求 (例如 99.99% 或更高可用性) 的關鍵任務工作負載，或只有容錯移轉至另一個區域才能滿足的嚴格作業持續性需求，可能適合採用多區域方法。不過，這些需求通常僅適用於企業工作負載產品組合的一小部分，這些產品組合的復原時間有限 (以分鐘或小時為單位)。除非應用程式需要幾分鐘或數小時的復原時間，否則在受影響的區域內等待應用程式的區域中斷進行修復可能是一種較好的方法，而且通常會與較低層級的工作負載保持一致。

在實施多區域架構之前，業務決策者和技術團隊應該就成本影響（包括營運和基礎架構成本驅動因素）保持一致。與單一區域方法相比，典型的多區域架構可能會增加兩倍的成本。雖然業務連續性有數種多區域模式，例如使用熱待命、暖待命和指示燈執行，但達到復原目標風險最低的模式將涉及執行熱待命，而且將使您的工作負載成本翻倍。

關鍵指引

- 應根據工作負載建立營運目標（例如 RTO 和 RPO）的可用性和持續性，並與業務和 IT 利益相關者保持一致。
- 可在單一區域內達成大多數營運目標的可用性和持續性。對於單一區域無法達成的目標，應該考慮多地區，並清楚瞭解成本、複雜性和收益之間的權衡。

多區域基礎 2：瞭解資料

對於多區域架構，管理資料是一個不平凡的問題。區域之間的地理距離會產生不可避免的延遲，這表現為跨區域複寫資料所需的時間。在可用性、資料一致性和使用多區域架構的工作負載中引入更高數量級的延遲是必要的。無論是使用非同步複寫還是同步複寫，您都需要修改應用程式，以處理複寫技術所強加的行為變更。由於數據一致性和延遲方面的挑戰，要採用專為單一區域設計的現有應用程序並使其成為多區域非常困難。瞭解特定工作負載的資料一致性需求和資料存取模式對於權衡取捨至關重要。

2a：了解數據一致性要求

CAP 定理提供有關資料一致性、可用性和網路分割區之間權衡的推理參考，其中一個工作負載只能同時滿足兩個分割區。根據定義的多區域包括區域之間的網路分割區，因此您必須在可用性和一致性之間進行選擇。

如果您選取跨區域的資料可用性，則在交易式寫入期間不會產生顯著延遲，因為在區域之間依賴非同步複寫已確認資料，進而在複寫完成之前降低區域間的一致性。使用非同步複寫時，當主要區域發生故障時，從主要區域寫入擱置複寫的可能性很高。這會導致最新資料在繼續複寫之前無法使用的情況，而且需要調解程序來處理未從發生中斷的區域複寫的執行中交易。

對於偏愛非同步複寫的工作負載，您可以使用提供非同步跨區域複寫的服務 (例如 Amazon Aurora 和 Amazon DynamoDB)。Amazon Aurora 全球資料庫和 Amazon DynamoDB 全域表都有預設的 Amazon CloudWatch 指標，可協助監控複寫延遲。

設計工作負載以利用事件驅動的架構是多區域策略的好處，因為這意味著工作負載可以採用非同步資料複寫，並透過重新播放事件來實現狀態的重建。由於串流和訊息服務會在單一區域中緩衝訊息承載資料，因此地區容錯移轉/容錯回復程序必須包含重新導向用戶端輸入資料流程的機制，以及協調儲存在發生中斷之區域中的執行中和/或未傳遞承載。

如果選取一致性，則在交易式寫入期間會同步複寫資料，因此會產生很大的延遲。同步寫入多個區域時，如果所有區域中的寫入都不成功，則可能會降低可用性，因為交易不會認可，因此需要重試。嘗試同步將資料寫入所有區域的重試，每次嘗試都會以延遲為代價完成。在某些時候，當重試已經用盡時，需要做出決定，要么完全失敗交易，從而降低可用性，或者只將交易提交到可用的區域，從而導致不一致。有一些法定成形技術，例如 Paxos，可以幫助同步複製和提交數據，但這需要大量的開發人員投資。

當寫入涉及跨多個區域的同步複寫以滿足嚴格的一致性需求時，寫入延遲會增加一定數量級。較高的寫入延遲不是通常可以在沒有重大變更的情況下重新安裝到應用程式中。理想情況下，首次設計應用程序時必須考慮到它。對於優先考量同步複製的多區域工作負載，AWS合作夥伴解決方案可提供協助。

2b：了解數據訪問模式

工作負載資料存取模式分為下列其中一種類型：讀取密集型或寫入密集型。瞭解特定工作負載的這項特性將引導選擇適當的多區域架構。

對於讀取密集型工作負載，例如完全唯讀的靜態內容，可以實現主動/主動式多區域架構，而不會顯著複雜。使用內容分發網路 (CDN) 在邊緣提供靜態內容，透過快取最接近最終使用者的內容來確保可用性；使用 Amazon 內的 Origin 容錯移轉等功能集 CloudFront可協助實現這一目標。另一個選項是在多個區域中部署無狀態運算，並使用 DNS 將使用者路由到最近的區域以讀取內容。可以使用具有地理位置路由策略的 53 號路線來實現這一目標。

對於讀取比例大於寫入比例的讀取密集型工作負載，可以使用讀取本機寫入全域策略。這需要將所有寫入到特定區域中的數據庫，並將數據異步複製到所有其他區域，並且可以在任何區域中完成讀取以實現此目的。此方法需要工作負載才能擁有最終的一致性，因為本機讀取可能會因為跨區域複寫寫入的延遲增加而過時。

Aurora Global Database 可協助在待命區域佈建僅能在本機處理所有讀取流量的僅供讀取複本，以及特定區域中的單一主要資料存放區來處理寫入。資料會以非同步方式從主要資料庫複製到待命資料庫 (僅供讀取複本)，如果您需要將作業容錯移轉至待命區域，則可將待命資料庫提升為主要資料庫。如果工作負載更適合非關聯式資料模型，DynamoDB 也可以用於此方法。再次，工作負載需要接受最終一致性，如果從一開始就不是為此而設計的，則可能需要重新寫入工作負載。

對於寫入密集型工作負載，應選取主要區域，並將容錯移轉至待命區域的功能設計到工作負載中。與主動/主動方法相比，主要/待機方法不太複雜。這是因為對於主動/主動架構而言，需要重寫工作負載以處理到區域的智慧路由、建立工作階段相似性、確保冪等交易，以及處理潛在衝突。

大多數正在尋找多區域恢復能力的工作負載不需要主動/主動方法。分片策略可以通過限制整個客戶群中減值的爆炸半徑來提供更高的彈性。如果您可以有效地分片用戶端基礎，則可以為每個碎片選取不同的主要區域。例如，如果您可以將用戶端分片，以便一半的用戶端與區域一對齊，而一半的用戶端與區域二對齊，將 Region 視為儲存格，則可以建立多區域儲存格方法，進而降低工作負載的衝擊半徑。

分片方法可以與主要/待命方法結合使用，以提供碎片的容錯移轉功能。經過測試的容錯移轉程序將需要設計到工作負載中，並且還需要設計資料調解程序，以確保容錯移轉後資料存放區的交易一致性。本 paper 稍後將詳細介紹這些內容。

關鍵指引

- 發生故障時，擱置複寫的寫入可能性很高不會送至待命區域。在繼續複寫之前，資料將無法使用 (假設非同步複寫)。

- 作為容錯移轉的一部分，將需要資料調解程序，以確保使用非同步複寫的資料存放區維護交易一致狀態。
- 當需要強大的一致性時，將需要修改工作負載，以容忍同步複製的資料存放區所需的延遲。

多區域基礎 3：瞭解您的工作負載相依性

特定工作負載在一個區域中可能有多個相依性，例如使用的AWS服務、內部相依性、協力廠商相依性、網路相依性、憑證、金鑰、密鑰和參數。為了確保工作負載在失敗情況下的操作，主要區域和待命區域之間應該沒有相依性；每個區域都應該能夠彼此獨立運作。為了實現這一目標，必須仔細檢查工作負載中的所有相依性，以確保它們在每個區域中都可用。這是必要的，因為主要區域中的故障不應該對待命區域產生影響。此外，當相依性處於降級狀態或完全無法使用時，工作負載如何運作的知識是必要的，因此可以設計解決方案以適當處理此問題。

3a：AWS服務項目

在設計多區域架構時，需要了解將要使用的特定AWS服務。第一個方面是了解該服務具有哪些功能才能實現多區域，以及是否必須設計解決方案才能實現多區域目標。例如，使用 Amazon Aurora 和 Amazon DynamoDB 時，您可以將資料以非同步方式複寫到待命區域的功能。任何AWS服務相依性都必須在要執行工作負載的所有區域中提供使用。為了確保將使用的服務在所需的地區可用，請查看 AWS 區域al 服務清單。

3b：內部和第三方相依性

對於工作負載具有的任何內部相依性，請確保工作負載將在其中運作的區域中可用。例如，如果工作負載是由許多微型服務組成，請瞭解構成商業能力的所有微服務。從那裡，確保所有這些微服務都部署在工作負載將運行出來的每個區域中。

不建議在工作負載內的微服務之間進行跨區域呼叫，並且應該保持區域隔離。 這是因為建立跨區域相依性會增加相關失敗的風險，這會否定您嘗試透過工作負載的隔離區域實作來達成的好處。內部部署相依性也可能是工作負載的一部分，因此必須瞭解這些整合的特性在主要區域變更時會如何變更。例如，如果待命區域位於離內部部署環境更遠，則延遲增加將產生負面影響。

瞭解軟體即服務 (SaaS) 解決方案、軟體開發套件 (SDK) 及其他協力廠商產品相依性，並能夠執行這些相依性降級或無法使用的情境，將提供更深入的瞭解系統鏈在不同故障模式下的運作和行為。這些相依性可能位於應用程式程式碼中，從使用 AWS Secrets Manager 或第三方保存庫解決方案 (例如 Hashicorp) 進行外部管理的密碼，再到依賴 IAM 身分中心以進行聯合登入的身份驗證系統。

在依賴關係方面具有冗餘可以幫助提高彈性。SaaS 解決方案或協力廠商相依性也可能使用與工作負載相同的主要AWS 區域項目。在這種情況下，您應該與廠商合作，以確定其彈性狀態是否符合工作負載的需求。

此外，請注意工作負載及其相依性之間的共同命運，例如第三方應用程式。如果在容錯移轉之後，次要區域中 (或從) 無法使用相依性，則工作負載可能無法完全復原。

3c：故障轉移機制

網域名稱系統 (DNS) 通常用作容錯移轉機制，將流量從主要區域轉移到待命區域。仔細檢閱和審查容錯移轉機制採用的所有相依性。例如，如果您的工作負載使用 Amazon Route 53，瞭解控制平面託管在 US-EAST-1 中，表示您正在依賴該特定區域中的控制平面。如果主要區域也是 US-EAS-1，則不建議將其作為容錯移轉機制的一部分。如果使用另一種容錯移轉機制，則需要深入瞭解無法如預期運作的任何案例。一旦建立了這種理解，計劃應變或發展一個新的機制，如果需要的話。請參閱使用 Amazon Route 53 建立災難復原機制，以了解可用於成功容錯移轉的方法。

如內部相依性一節所述，屬於商務功能一部分的所有微服務都必須在部署工作負載的每個區域中提供使用。作為容錯移轉策略的一部分，業務功能需要同時進行容錯移轉，以消除跨區域呼叫的機會。或者，如果微服務容錯移轉獨立，這會導致微服務可能進行跨區域呼叫的不良行為的可能性，這會導致延遲，並可能導致工作負載在用戶端逾時的情況下無法使用。

3d：配置依賴關係

憑證、金鑰、機密和參數是針對多區域進行設計時所需的相依性分析的一部分。如果可能的話，最好在每個區域內本地化這些元件，這樣它們就不會因為這些相依性在區域之間有共同命運。對於憑證，憑證的到期應該會有所不同，如果可能的話，在每個區域中，以避免憑證過期 (設定為預先通知的警示) 影響多個區域的情況。

加密金鑰和密碼也應該是特定於區域的。如此一來，如果金鑰或機密的輪換發生錯誤，則影響僅限於特定區域。

最後，任何工作負載參數都應儲存在本機，以便在特定區域擷取工作負載。

關鍵指引

- 多區域架構受益於區域之間的實體和邏輯區隔。在應用程式層引入跨區域相依性可打破這項優勢。避免這種依賴關係。
- 容錯移轉控制應在主要區域上沒有相依性的情況下運作。
- 需要完成業務功能的協調容錯移轉，以消除跨區域呼叫延遲增加和依賴性的可能性。

多區域基礎 4：營運準備就緒

操作多區域工作負載是一項複雜的工作，會面臨多地區特有的營運挑戰。其中包括AWS 帳戶管理、改造部署程序、建立多區域觀察策略、建立和測試容錯移轉和容錯回復 Runbook，然後管理成本。作業準備檢閱 (ORR) 可協助團隊準備工作負載以進行生產，無論是在單一區域執行還是跨多個區域執行。

4a：AWS 帳戶管理層

若要跨區域部署工作負載AWS 區域，請確定帳戶內的所有AWS服務配額都是同位檢查。首先，了解屬於架構一部分的所有AWS服務，查看待命區域中的計劃使用情況，然後將它們與目前的使用情況進行比較。在某些情況下，如果之前未使用過待命區域，您可以參考預設服務配額以瞭解起點。然後，在將使用的所有服務中，使用 Service Quotas 控制台（需要登錄）或 API 請求增加配額。

AWSIdentity and Access Management (IAM) 角色必須在每個區域中設定，以確保操作員、自動化工具和AWS服務具有待命區域內資源的適當許可。區域隔離角色實現了我們對多區域架構所追求的區域隔離。在使用待命區域上線之前，請確保這些權限已就位。

4b：部署做法

使用多區域功能，將工作負載部署到多個區域可能非常複雜。 AWS CloudFormation有助於將基礎架構部署到單個或多個區域，並且可以根據您的需求量身定制。 AWS CodePipeline有助於提供幾乎持續的整合/持續交付 (CI/CD) 管線，該管線具有跨區域動作，可讓您部署至與管道所在區域不同的區域。這與藍/綠等強大的部署策略相結合，可實現最低至零的停機時間部署。

不過，當應用程式或資料的狀態未外部化至永久性存放區時，可設定狀態功能的部署可能會比較複雜。在這些情況下，請仔細調整部署程序以符合您的需求。設計部署管道和程序，一次在一個區域部署，而不是同時部署多個區域。這減少了區域之間相關故障的機會。若要了解 Amazon 用於自動化軟體部署的技巧，請閱讀 Builder Library 文章自動化安全、免動手部署。

4c：可觀察性

針對多區域進行設計時，請考慮如何監控每個區域中所有元件的健康狀況，以獲得區域健康狀況的全面檢視。這可能包括監視複寫延遲的指標，這不是單一區域工作負載的考量。

建置多區域架構時，也請考慮觀察待命區域的工作負載效能。 這包括從待命區域運行健康檢查和 Canary（綜合測試），從而提供主要區域的健康狀況的外部視圖。此外，您可以使用 Amazon 網 CloudWatch 際網路監視器從最終使用者的角度瞭解外部網路的狀態和工作負載的效能。同樣，主要區

域應具有相同的可觀測性來監視待命區域。這些 Canary 應監視客戶體驗指標，以取得工作負載的整體健康狀況。這是必要的，因為如果主要區域出現問題，則主要區域的可觀察性可能會受損，並且會影響評估工作負載健康狀況的能力。

在這種情況下，在該區域之外進行觀察可以提供見解。這些指標應匯總到每個區域中可用的儀表板中，並在每個區域中建立警示。由於 Amazon CloudWatch 是區域服務，因此必須在兩個區域使用這些服務。此監視資料將用於從主要區域到待命區域進行容錯移轉的呼叫。

4d：過程，程序和測試

回答這個問題的最佳時機：「何時應該容錯移轉？」 很久之前，你需要。包括人員、流程和技術的業務連續性計劃都應在問題發生之前定義良好，並定期進行測試。決定恢復決策框架。如果有適當實踐的復原程序，而且已充分瞭解復原的時間，則可以選擇透過容錯移轉來啟動復原程序符合 RTO 目標的復原程序的時間點。此時間點可能是在主要區域中的應用程式發生問題之後立即發生，或者可能會進一步發生區域中應用程式中的復原選項已用盡的事件，而且現在應該啟動容錯移轉以符合 RTO。

雖然容錯移轉動作本身應該是 100% 自動化，啟動容錯移轉的決定應由人員 (通常是組織中的少數預先決定的個人) 來決定。此外，決定容錯移轉的準則也必須與組織一起清楚定義並全域瞭解。這些程序可以使用 AWSSystem Manager Runbook 來定義和完成，這樣可以完全 end-to-end 自動化，並確保在測試和容錯移轉期間執行程序的一致性。

這些 Runbook 應在主要和待命區域中提供，以啟動容錯移轉或容錯回復程序。一旦這種自動化到位，應定期測試節奏定義和遵循。這確保了當有一個實際的事件時，響應是在一個明確定義的，實踐的過程中運行，該組織有信心。同樣重要的是要牢記資料調解程序的既定公差。確認已建立的 RPO/RTO 要求符合提議的程序。

4e：成本和複雜性

多區域架構的成本影響是由更高的基礎架構使用量、營運開銷和資源時間所帶動。如前所述，待命區域中的基礎架構成本與預先佈建時主要區域的基礎架構成本相似，因此成本是兩倍。佈建容量，使其足以執行日常作業，但仍保留足夠的緩衝區容量以容忍需求峰值，並在每個區域設定相同的限制。

此外，如果您採用主動-主動式架構，可能需要應用程式層級的變更才能在多區域架構中順利執行，因此設計和操作可能需要耗費大量時間和資源。組織至少需要花時間了解每個區域中的技術和業務相依性，以及設計容錯移轉和容錯回復程序。

團隊還應該通過正常的故障轉移和故障回復練習，以使用將在事件期間使用的 Runbook 感到舒適。雖然這些練習對於從多地區投資中獲得預期結果非常重要且至關重要，但這些練習代表了機會成本，並且需要花費時間和資源遠離其他活動。

關鍵指引

- AWS在工作負載將運作的所有區域中，都需要檢查服務配額並同步檢查。
- 部署程序應該一次以一個區域為目標，而不是同時針對多個區域。
- 需要監視複寫延遲等其他指標，並且特定於多地區案例。
- 將工作負載的監控延伸到主要區域之外。應監控每個區域的客戶體驗指標，並從執行工作負載的每個區域之外進行測量。
- 容錯移轉和容錯回復需要定期測試。確保在測試和即時事件期間使用的容錯移轉和容錯回復程序實作單一 Runbook。用於測試和現場活動的手冊不能不同。

結論

本白皮書討論了多區域的常見使用案例、如何實作多區域架構的基礎知識，以及此方法的含意。這些基本原理可以應用於任何工作負載，並用作框架，以幫助決策多區域架構是否適合特定業務。

貢獻者

本文件的貢獻者包括：

技術貢獻者：

- 約翰·福爾門托，主要解決方案架構師，AWS多區域團隊

編輯貢獻者：

- 李西·劉易斯，高級經理，產品行銷

深入閱讀

如需其他資訊，請參閱：

- 進階異地同步備份復原模式 (AWS白皮書)
- 可靠性支柱-AWS Well-Architected 的框架
- 可用性和超越：了解並提高分散式系統的復原能力 AWS（AWS白皮書）
- AWS故障隔離界限 (AWS白皮書)

文件修訂

若要收到有關此白皮書更新的通知，請訂閱 RSS 摘要。

變更	描述	日期
文件已發佈	第一次出版。	2022 年 12 月 20 日

注意

客戶有責任對本文件中的資訊進行自己的獨立評定。本文件：(a) 僅供參考之用；(b) 代表目前 AWS 產品供應與實務，如有變更恕不另行通知；以及 (c) 不構成 AWS 及其附屬公司、供應商或授權人的任何承諾或保證。AWS 產品或服務均以「原樣」提供，不作任何形式的明示或暗示的保證、陳述或條件。AWS 對其客戶的責任與義務應由 AWS 協議管轄，本文並非 AWS 與其客戶之間的任何協議的一部分，也並非上述協議的修改。

© 2022 Amazon Web Services, Inc. 或其附屬公司。保留所有權利。

AWS 詞彙表

如需最新的 AWS 術語，請參閱《AWS 詞彙表 參考》中的 AWS 詞彙表。

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。