使用者指南

Amazon S3 on Outposts

API 版本 2006-03-01

Amazon S3 on Outposts: 使用者指南

Table of Contents

什麼是 Amazon S3 on Outposts？

AWS Outposts 是一種全受管服務，可為幾乎任何資料中心、主機代管空間或內部部署設施提供相同的 AWS 基礎設施、 AWS 服務、APIs 和工具，以獲得真正一致的混合體驗。 AWS Outposts 非常適合需要低延遲存取內部部署系統、本機資料處理、資料駐留，以及遷移具有本機系統相互依存性之應用程式的工作負載。如需詳細資訊，請參閱《 AWS Outposts使用者指南》中的什麼是AWS Outposts ？。

使用 Amazon S3 on Outposts，您可以在 Outposts 上建立 S3 儲存貯體並輕鬆存放和擷取內部部署物件。S3 on Outposts 提供一個全新的儲存類別，即 OUTPOSTS，使用 Amazon S3 API，目的是在您的 Outposts 上的多個裝置和伺服器上以持久、備援的方式存放資料。您可以使用存取點和透過虛擬私有雲端 (VPC) 的端點連線，與您的 Outposts 儲存貯體進行通訊。

就像在 Amazon S3 一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過 AWS Management Console、 AWS Command Line Interface (AWS CLI)、 AWS SDKs 或 REST API 使用 S3 on Outposts。

- S3 on Outposts 如何工作
- S3 on Outposts 功能
- 相關服務
- 存取 S3 on Outposts
- 支付 S3 on outposts
- 後續步驟

S3 on Outposts 如何工作

S3 on Outposts 是將資料當做物件存放在您的 Outpost 儲存貯體中的物件儲存服務。物件是一個資料檔案和任何描述該檔案的中繼資料。儲存貯體是物件的容器。

若要將資料存放在 S3 on Outposts 中，您首先要建立儲存貯體。建立儲存貯體時，您需要指定儲存貯體名稱和將保存儲存貯體的 Outpost。若要存取 S3 on SOutposts 儲存貯體並執行物件操作，接下來需要建立並設定一個存取點。您還必須建立端點，以將請求路由到存取點。

存取點可簡化將資料存放在 S3 中的任何 AWS 服務 或客戶應用程式的資料存取。存取點為連接到儲存貯體的指定網路端點，這些端點可用於執行物件操作，例如 `GetObject` 和 `PutObject`。每個存取點都有不同的許可和網路控制。

您可以使用 、、 AWS SDKs 或 REST API 來建立和管理 S3 on Outposts 儲存貯體 AWS Management Console AWS CLI、存取點和端點。若要上傳和管理 S3 on Outposts 儲存貯體中的物件，您可以使用 、 AWS CLI AWS SDKs或 REST API。

區域

在 AWS Outposts 佈建期間，您或 AWS 會建立服務連結連線，將您的 Outpost 連線至您選擇的 AWS 區域 或 Outposts 主區域，以進行儲存貯體操作和遙測。Outpost 依賴於與父節點之間的連接 AWS 區域。Outposts 機架不適用於斷開連接的操作或連接受限制的環境。如需詳細資訊，請參閱《AWS Outposts 使用者指南》中的Outpost 連線 AWS 區域。

儲存貯體

儲存貯體是 S3 on Outposts 中用於存放物件的容器。您可以在儲存貯體中存放任意數目的物件，並且每個 Outpost 中的每個帳戶最多可有 100 個儲存貯體。

建立儲存貯體時，請輸入儲存貯體名稱並選擇儲存貯體將駐留的 Outpost。建立儲存貯體後，便無法變更儲存貯體名稱或移動儲存貯體到不同 Outpost。儲存貯體名稱必須遵循Amazon S3 儲存貯體命名規則。在 S3 on Outposts 中，儲存貯體名稱對 Outpost 和 是唯一的 AWS 帳戶。S3 on Outposts 儲存貯體需要 `outpost-id`、`account-id` 和儲存貯體名稱來識別。

下列範例顯示了 S3 on Outposts 儲存貯體 的 Amazon Resource Name (ARN) 格式。ARN 由您的 Outpost 所在區域、您的 Outpost 帳戶、Outpost ID 和儲存貯體名稱組成。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

每個物件都包含在儲存貯體中。您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。針對物件操作指定儲存貯體時，您可以使用存取點 ARN 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

下列範例顯示了 S3 on Outposts 的存取點 ARN 格式，其中包含了 `outpost-id`、`account-id` 和存取點名稱：

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

如需儲存貯體的詳細資訊，請參閱使用 S3 on Outposts 儲存貯體。

物件

物件是存放在 S3 on Outposts 中的基本實體。物件是由物件資料與中繼資料構成。中繼資料是一組成對的名稱與數值，會說明該物件。其中包含一些預設中繼資料 (如上次修改日期) 以及標準 HTTP 中繼資料 (如 `Content-Type`)。您也可以在存放物件時指定自訂中繼資料。在儲存貯體中，每個物件都是由金鑰 (名稱) 來唯一識別。

對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

金鑰

物件金鑰 (或金鑰名稱) 是儲存貯體內的物件的唯一識別碼。儲存貯體中的每個物件只能有一個金鑰。儲存貯體和物件金鑰的組合唯一識別每個物件。

下列範例顯示 S3 on Outposts 物件的 ARN 格式，其中包含 Outpost 所在區域的 AWS 區域 程式碼、AWS 帳戶 ID、Outpost ID、儲存貯體名稱和物件金鑰：

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket1/object/myobject
```

如需物件金鑰的詳細資訊，請參閱使用 S3 on Outposts 物件。

S3 版本控制

您可以在 Outposts 上使用 S3 版本控制，以在相同的儲存貯體中保留物件的多個變體。使用 S3 版本控制功能，您即可保留、擷取和還原在儲存貯體中所存放每個物件的各個版本。S3 版本控制可協助您從意外的使用者動作和應用程式失敗中復原。

如需詳細資訊，請參閱針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制。

版本 ID

當您在儲存貯體中啟用 S3 版本控制時，S3 on Outposts 會針對每個新增至儲存貯體的物件產生唯一的版本 ID。啟用版本控制時已存在於儲存貯體中的物件的版本 ID 為 `null`。如果您使用其他操作修改這些 (或任何其他) 物件，例如 PutObject，新物件會取得唯一的版本 ID。

如需詳細資訊，請參閱針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制。

儲存類別和加密

S3 on Outposts 提供新的儲存類別 S3 Outposts (OUTPOSTS)。S3 Outposts 儲存方案適用於只存放在 AWS Outposts儲存貯體中的物件。如果您嘗試與 S3 on Outposts 一起使用其他 S3 儲存方案，S3 on Outposts 會返回 `InvalidStorageClass` 錯誤。

根據預設，物件存放在 S3 Outposts (OUTPOSTS) 儲存體方案中的物件一律使用伺服器端加密與 Amazon S3 受管加密金鑰 (SSE-S3) 進行加密。如需詳細資訊，請參閱S3 on Outposts 中的資料加密。

儲存貯體政策

儲存貯體政策是資源型 AWS Identity and Access Management (IAM) 政策，可用來授予儲存貯體及其物件的存取許可。只有儲存貯體擁有者可建立政策與儲存貯體的關聯。連接到儲存貯體的許可會套用至儲存貯體擁有者帳戶擁有的所有儲存貯體物件。儲存貯體政策的大小限制為 20 KB。

儲存貯體政策使用 AWS標準的以 JSON 為基礎的 IAM 政策語言。您可以使用儲存貯體政策來新增或拒絕儲存貯體中物件的許可。儲存貯體政策允許或拒絕請求以政策中的元素為基礎。這些元素可以包括請求的申請者、S3 on Outposts 動作、資源以及其他方面或條件 (例如，用來傳送要求的 IP 地址)。例如，您可以建立儲存貯體政策，授予跨帳戶許可，以將物件上傳至 S3 on Outposts 儲存貯體，同時確保儲存貯體擁有者可完全控制上傳物件。

在儲存貯體政策中，您可以使用 ARN 格式的萬用字元 (*) 和其他值將許可授予物件子集。例如，您可以控制對以常用字首開頭或以給定的副檔名結束的一組物件存取權，例如 `.html`。

S3 on Outposts 存取點

S3 on Outposts 存取點是含有專用存取政策的命名網路端點，其中說明了如何使用該端點存取資料。存取點針對 S3 on Outposts 中的共用資料集，簡化了對大規模資料存取的管理。存取點為連接到儲存貯體，您可以使用這些端點來執行 S3 物件操作，例如 `GetObject` 和 `PutObject`。

針對物件操作指定儲存貯體時，您可以使用存取點 ARN 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

存取點有 S3 on Outposts 對於透過該存取點進行的任何請求所套用的不同許可和網路控制。每個存取點都會強制執行自訂的存取點政策，該政策可結合附加至基礎儲存貯體的儲存貯體政策運作。

如需詳細資訊，請參閱存取 S3 on Outposts 儲存貯體和物件。

S3 on Outposts 功能

存取管理

S3 on Outposts 提供稽核和管理對儲存貯體和物件的存取的功能。根據預設，S3 on Outposts 儲存貯體與物件皆為私有。您只能存取您建立的 S3 on Outposts 資源。

若要授予可支援特定使用案例的精密資源使用權限，或稽核 S3 on Outposts 資源的許可，您可以使用下列功能。

- S3 區塊公有存取 – 儲存貯體與物件的區塊公有存取。對於 Outposts 上的儲存貯體，預設情況下始終啟用封鎖公開存取。
- AWS Identity and Access Management (IAM) – IAM 是一種 Web 服務，可協助您安全地控制對 AWS 資源的存取，包括 S3 on Outposts 資源。使用 IAM，您可以集中管理控制使用者可以存取哪些 AWS 資源的許可。您可以使用 IAM 來控制能通過身分驗證 (登入) 和授權使用資源的 (具有許可) 的人員。
- S3 on Outposts 存取點 - 管理針對 S3 on Outposts 中共用資料集的資料存取。存取點以專屬存取政策命名網路端點。存取點與儲存貯體相關聯，可用於執行物件操作，例如 `GetObject` 和 `PutObject`。
- 儲存貯體政策 – 使用以 IAM 為基礎的政策語言，為 S3 儲存貯體及其中的物件設定以資源為基礎的許可。
- AWS Resource Access Manager (AWS RAM) – 在您的組織或組織單位 (OUs) AWS 帳戶之間安全地共用 S3 on Outposts 容量 AWS Organizations。

儲存記錄和監控

S3 on Outposts 提供記錄和監控工具，您可以使用這些工具來監控和控制 S3 on Outposts 資源的使用方式。如需詳細資訊，請參閱監控工具。

- 適用於 S3 on Outposts 的 Amazon CloudWatch 指標 - 追蹤資源的運作狀態並瞭解您的容量可用性。
- 適用於 S3 on Outposts 的 Amazon CloudWatch Events 事件 - 為 S3 on Outposts API 事件建立規則，以便通過所有受支持的 CloudWatch Events 目標接收通知，包括 Amazon Simple Queue Service (Amazon SQS)、Amazon Simple Notification Service (Amazon SNS) 和 AWS Lambda。
- AWS CloudTrail S3 on Outposts 的 日誌 – 在 S3 on Outposts AWS 服務 中記錄使用者、角色或 採取的動作。CloudTrail 日誌為您提供 S3 儲存貯體層級和物件層級操作的詳細 API 追蹤。

高度的一致性

S3 on Outposts 為 S3 on Outposts 儲存貯體中物件的 PUT 和 DELETE 請求提供強大的read-after-write一致性 AWS 區域。這一行為適用於新物件的寫入，和覆寫現有物件的 PUT 請求與 DELETE 請求。此外，S3 on Outposts 物件標籤和物件中繼資料 (例如 HEAD 物件) 高度一致。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的 Amazon S3 資料一致性模式。

相關服務

將資料載入至 S3 on Outposts 之後，您可以搭配其他 AWS 服務使用資料。以下為您可能最常使用的服務：

- Amazon Elastic Compute Cloud (Amazon EC2) – 在 AWS 雲端中提供安全且可擴展的運算容量。使用 Amazon EC2 可減少前期所需的硬體投資，讓您更快速開發並部署應用程式。您可使用 Amazon EC2 按需要啟動任意數量的虛擬伺服器，設定安全性和聯網功能以及管理儲存。
- Outpost 上的 Amazon Elastic Block Store (Amazon EBS) - 使用 Amazon EBS 本機快照 將 Outpost 上的磁碟區快照以本機方式儲存在 Outpost 上的 S3 on Outposts 本身。
- Amazon Relational Database Service (Amazon RDS) - 使用 Amazon RDS 本地備份將您的 Amazon RDS 備份本地存儲在您的 Outpost 上。
- AWS DataSync – 自動在 Outpost 和 之間傳輸資料 AWS 區域，選擇要傳輸的內容、何時傳輸，以及要使用的網路頻寬。S3 on Outposts 已與 整合 AWS DataSync。對於需要高輸送量本機處理的內部部署應用程式，S3 on Outposts 提供內部部署物件儲存體，以盡量減少因網路變化而產生的資料傳輸和緩衝區，同時還能讓您輕鬆地在 Outpost 與 AWS 區域之間傳輸資料。

存取 S3 on Outposts

您可以透過以下任何方式來使用 S3 on Outposts：

AWS Management Console

主控台是可用來管理 S3 on Outposts 和 AWS 資源的 Web 型使用者介面。如果您已註冊 AWS 帳戶，您可以登入 AWS Management Console 並從首頁選擇 S3，以存取 S3 AWS Management Console on Outposts。 S3 然後，從左側導覽窗格中選擇 Outposts buckets (Outposts 儲存貯體)。

AWS Command Line Interface

您可以使用 AWS 命令列工具，在系統的命令列發出命令或建置指令碼來執行 AWS （ 包括 S3) 任務。

AWS Command Line Interface (AWS CLI) 為廣泛的 提供命令 AWS 服務。Windows、macOS 和 Linux AWS CLI 支援 。若要開始使用，請參閱《AWS Command Line Interface 使用者指南》。如需可與 S3 on Outposts 搭配使用之命令的詳細資訊，請參閱 《AWS CLI 命令參考》中的 s3api、s3control 和 s3outposts。

AWS SDKs

AWS 提供SDKs開發套件 （軟體開發套件），其中包含適用於各種程式設計語言和平台 (Java、Python、Ruby、.NET、iOS、Android 等） 的程式庫和範本程式碼。 AWS SDKs 提供便捷的方式，可建立 S3 on Outposts 和 的程式設計存取 AWS。由於 S3 on Outposts 使用與 Amazon S3 相同的軟件開發工具包，所以 S3 on Outposts 使用相同的 S3 API、自動化和工具提供一致的體驗。

S3 on Outposts 是 REST 服務。您可以使用 AWS 開發套件程式庫 (其會包裝基礎 REST API)，傳送請求至 S3 on Outposts，從而簡化程式設計任務。例如，開發套件會負責的工作諸如計算簽章、以密碼演算法簽署請求、管理錯誤以及自動重試請求。如需有關 AWS SDKs 的資訊，包括如何下載和安裝它們，請參閱建置工具 AWS。

支付 S3 on outposts

您可以購買各種 AWS Outposts 機架組態，結合 Amazon EC2 執行個體類型、Amazon EBS 一般用途固態硬碟 (SSD) 磁碟區 (gp2) 和 S3 on Outposts。定價包括運輸、安裝、基礎設施服務維護 、 以及軟體修補程式和升級。

如需詳細資訊，請參閱 AWS Outposts rack 定價。

後續步驟

如需有關使用 S3 on Outposts 的詳細資訊，請參閱下列主題：

- 設定 Outpost
- Amazon S3 on Outposts 與 Amazon S3 有何不同？
- Amazon S3 on Outposts 入門
- 適用於 S3 on Outposts 的網路
- 使用 S3 on Outposts 儲存貯體
- 使用 S3 on Outposts 物件
- S3 on Outposts 中的安全性
- 管理 S3 on Outposts 儲存貯體

- 使用 Amazon S3 on Outposts 進行開發

設定 Outpost

若要開始使用 Amazon S3 on Outposts，您需要在設施中部署有 Amazon S3 容量的 Outpost。如需訂購 Outpost 和 S3 容量選項的詳細資訊，請參閱 AWS Outposts。若要檢查您的 Outposts 是否有 S3 容量，您可以使用 ListOutpostsWithS3 API 呼叫。有關規範以及要瞭解 S3 on Outposts 與 Amazon S3 有何不同，請參閱 Amazon S3 on Outposts 與 Amazon S3 有何不同？。

如需詳細資訊，請參閱下列主題。

主題

- 訂購新 Outpost

訂購新 Outpost

如果您需要訂購含 S3 容量的新 Outpost，請參閱 AWS Outposts 定價，以瞭解 Amazon Elastic Compute Cloud (Amazon EC2)、Amazon Elastic Block Store (Amazon EBS) 和 Amazon S3 的容量選項。

選取您的組態之後，請依照 AWS Outposts 使用者指南中的建立 Outpost 和訂購 Outpost 容量的步驟操作。

Amazon S3 on Outposts 與 Amazon S3 有何不同？

Amazon S3 on Outposts 會將物件儲存體交付至您的內部部署 AWS Outposts 環境。使用 S3 on Outposts 可透過將資料保留在靠近內部部署應用程式的位置，協助您滿足本機處理、資料駐留、和嚴苛的效能需求。由於 S3 on Outposts 使用 Amazon S3 APIs 和功能，因此可讓您輕鬆儲存、保護、標記、報告和控制對 Outposts 上資料的存取，並將 AWS 基礎設施擴展至內部部署設施，以獲得一致的混合體驗。

如需 S3 on Outposts 特點的詳細資訊，請參閱下列主題。

主題

- S3 on Outposts 規格
- S3 on Outposts 支援的 API 操作
- Amazon S3 AWS CLI S3 命令
- S3 on Outposts 不支援的 Amazon S3 功能
- S3 on Outposts 網路需求

S3 on Outposts 規格

- Outposts 儲存貯體大小上限為 50 TB。
- 每個 AWS 帳戶的 Outposts 儲存貯體數目上限為 100。
- 只能使用存取點和端點來存取 Outposts 儲存貯體。
- 每個 Outpost 儲存貯體的存取點數目上限為 10。
- 存取點政策的大小限制為 20 KB。
- Outpost 擁有者可以使用 在 中管理組織內 AWS Organizations 的存取 AWS Resource Access Manager。所有需要存取 Outpost 的帳戶必需在與 AWS Organizations中的擁有者帳戶相同的組織內。
- S3 on Outpost 儲存貯體擁有者帳戶一律是儲存貯體中所有物件的擁有者。
- S3 on Outpost 儲存貯體擁有者帳戶只能對儲存貯體執行作業。
- 物件大小限制與 Amazon S3 一致。
- 所有儲存在 S3 on Outposts 上的物件都儲存在 OUTPOSTS 儲存類別中。

- 存放在 OUTPOSTS 儲存體類別中的所有物件預設為使用伺服器端加密與 Amazon S3 受管加密金鑰 (SSE-S3) 進行儲存。您也可以明確選擇使用伺服器端加密與客戶提供的加密金鑰 (SSE-C) 來存放物件。
- 如果無足夠的空間可以在您的 Outpost 上儲存物件，API 將傳回容量不足例外狀況 (ICE)。

S3 on Outposts 支援的 API 操作

如需 S3 on Outposts 支援的 API 操作清單，請參閱 Amazon S3 on Outposts API 操作。

Amazon S3 AWS CLI S3 命令

Amazon S3 AWS CLI on Outposts 目前支援下列 Amazon S3 命令。如需詳細資訊，請參閱《命令參考》中的可用 AWS CLI 命令。

- `cp`、 `mv`和 位於相同的儲存Outposts貯`sync`體內，或在本機環境和 Outposts 儲存貯體之間。
- `ls`
- `presign`
- `rm`

S3 on Outposts 不支援的 Amazon S3 功能

Amazon S3 on Outposts 目前不支援下列 Amazon S3 功能。任何嘗試使用它們操作都會遭拒絕。

- 條件式請求
- 存取控制清單 (ACL)
- 跨來源資源分享 (CORS)
- S3 批次操作
- S3 庫存報告
- 變更預設儲存貯體加密
- 公有儲存貯體
- 多重要素驗證 (MFA) 刪除
- S3 生命週期轉移 (物件刪除和停用不完整的分段上傳除外)
- S3 物件鎖定法務保存

- 物件鎖定保留
- 伺服器端加密搭配 AWS Key Management Service (AWS KMS) 金鑰 (SSE-KMS)
- S3 複寫時間控制 (S3 RTC)
- Amazon CloudWatch 請求指標
- 指標組態
- Transfer Acceleration
- S3 事件通知
- 申請者付款儲存貯體
- S3 Select
- AWS Lambda 事件
- Server access logging (伺服器存取記錄日誌)
- HTTP POST 請求
- SOAP
- 網站存取

S3 on Outposts 網路需求

- 若要將請求路由至 S3 on Outpost 存取點，您必須建立和設定 S3 on Outposts 端點。下列限制適用於 S3 on Outpost 的端點：
  - Outpost 上的每個 Virtual Private Cloud (VPC) 可以有一個相關聯的端點，而且每個 Outpost 最多可以有 100 端點。
  - 您可以將多個存取點對應至同一個端點。
  - 您只能將端點新增至具有以下 CIDR 範圍子空間中 CIDR 區塊的 VPC：
    - 10.0.0.0/8
    - 172.16.0.0/12
    - 192.168.0.0/16
- 您只能從具有非重疊 CIDR 區塊的 VPC 建立 Outpost 端點。
- 您只能從其 Outpost 子網路內建立端點。
- 您用於建立端點的子網路必須包含可供 S3 on Outpost 使用的四個 IP 地址。
- 如果您指定客戶擁有的 IP 地址集區 (CoIP 集區)，它必須包含可供 S3 on Outposts 使用的四個 IP 地址。

- 每個 VPC 的每個 Outpost 只能建立一個端點。

Amazon S3 on Outposts 入門

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (OUTPOSTS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。

透過 Amazon S3 on Outposts，您可以在 上使用 Amazon S3 APIs 和功能，例如物件儲存、存取政策、加密和標記， AWS Outposts 就像在 Amazon S3 上一樣。如需 S3 on Outposts 的資訊，請參閱 什麼是 Amazon S3 on Outposts？。

主題

- 開始使用 AWS Management Console
- 開始使用適用於 Java 的 AWS CLI 和 開發套件

開始使用 AWS Management Console

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (OUTPOSTS)，其使用 Amazon S3 APIs，旨在在您的多個裝置和伺服器上以持久且備援的方式存放資料 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、SDK 或 REST API 使用 S3 on Outposts。 AWS SDKs 如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

若要開始透過主控台使用 S3 on Outposts，請參閱下列主題。若要開始使用 AWS CLI 或 適用於 Java 的 AWS SDK，請參閱 開始使用適用於 Java 的 AWS CLI 和 開發套件。

主題

- 建立儲存貯體、存取點和端點
- 後續步驟

建立儲存貯體、存取點和端點

下列程序示範如何在 S3 on Outposts 中建立第一個儲存貯體。當您使用主控台建立儲存貯體時，您還會建立一個存取點和與儲存貯體關聯的端點，以便您可以立即開始在儲存貯體中存儲物件。

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台開啟 https：//https://console.aws.amazon.com/s3/。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇 Create Outposts bucket (建立 Outposts 儲存貯體)。
4. 在 Bucket name (儲存貯體名稱) 中，為儲存貯體輸入符合網域名稱系統 (DNS) 規範的名稱。

   儲存貯體名稱必須；

   - 在 AWS 帳戶、Outpost 和 Outpost 所在的 中是唯一 AWS 區域 的。
   - 長度必須介於 3–63 個字元之間。
   - 不含大寫字元。
   - 以小寫字母或數字開頭。

   建立儲存貯體後，便無法變更其名稱。如需為儲存貯體命名的相關資訊，請參閱《Amazon S3 使用者指南》中的一般用途儲存貯體命名規則。

   > **Important**
   >
   > 避免在儲存貯體名稱中包含敏感資訊，例如帳戶號碼。在指向儲存貯體中之物件的 URL 中，會顯示儲存貯體名稱。

5. 在 Outpost 中，選擇您要儲存貯體駐留的 Outpost。
6. 在 Bucket Versioning (儲存貯體版本控制) 下，將 S3 on Outposts 儲存貯體的 S3 版本控制狀態設定為下列其中一個選項：

   - Disable (停用) (預設) - 儲存貯體會保留未版本控制的狀態。
   - Enable (啟用) - 針對儲存貯體中的物件啟用 S3 版本控制。所有新增至儲存貯體的物件都會收到唯一的版本 ID。

   如需 S3 版本控制的詳細資訊，請參閱「針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制」。
7. (選擇性) 新增要與 Outposts 儲存貯體建立關聯的任何 optional tags (選用標籤)。您可以使用標籤來追蹤個別專案或一組專案的條件，或者使用成本分配標籤來標示儲存貯體。

存放在 Outposts 儲存貯體中的所有物件預設為使用伺服器端加密與 Amazon S3 受管加密金鑰 (SSE-S3) 進行儲存。您也可以明確選擇使用伺服器端加密與客戶提供的加密金鑰 (SSE-C) 來存放物件。若要變更加密類型，您必須使用 REST API、 AWS Command Line Interface (AWS CLI) 或 AWS SDKs。

8. 在 Outposts access point settings (Outposts 存取點設定) 區段中，輸入存取點名稱。

   S3 on Outposts 存取點針對 S3 on Outposts 中的共用資料集，簡化了對大規模資料存取的管理。存取點為連接到 Outposts 儲存貯體的具名網路端點，您可以使用這些端點來執行 S3 物件操作。如需詳細資訊，請參閱存取點。

   存取點名稱在此區域和 Outpost 的帳戶中必須是唯一的，並且符合存取點約束與限制。

9. 對此 Amazon S3 on Outposts 存取點選擇 VPC。

   如果您沒有 VPC，請選擇 Create VPC (建立 VPC)。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的建立僅限虛擬私有雲端 (VPC) 使用的存取點。

   Virtual Private Cloud (VPC) 可讓您將 AWS 資源啟動到您定義的虛擬網路。這個虛擬網路與您在資料中心中操作的傳統網路非常相似，且具備使用 可擴展基礎設施的優勢 AWS

10. (對於現有 VPC 可選) 請為您的端點選擇Endpoint subnet (端點子網路)。

    子網路是您的 VPC 中的 IP 地址範圍。如果您沒有想要的子網路，請選擇 Create subnet (建立子網路)。如需詳細資訊，請參閱適用於 S3 on Outposts 的網路。

11. (對於現有 VPC 可選) 請為您的端點選擇Endpoint security group (端點安全群組)。

    安全群組做為虛擬防火牆以控制傳入及傳出流量。

12. (對於現有 VPC 可選) 請選擇 Endpoint access type (端點存取類型) :

    - 私有 - 與 VPC 一起使用。
    - 客戶擁有的 IP - 與內部部署網路內的客戶擁有的 IP 地址集區 (CoIP 集區) 一起使用。

13. (選用) 指定Outpost 存取點政策。主控台會自動顯示存取點的 Amazon Resource Name (ARN)，您可以在政策中使用該名稱。

14. 選擇 Create Outposts bucket (建立 Outposts 儲存貯體)。

> **Note**
>
> 建立您的 Outpost 端點以及您的儲存貯體準備就緒可以使用，可能需要 5 分鐘。若要設定其他儲存貯體設定，請選擇 View details (檢視詳細資訊)

後續步驟

對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

建立 S3 on Outposts 儲存貯體、存取點和端點之後，您可以使用適用於 Java 的 AWS CLI 或 SDK 將物件上傳至儲存貯體。如需詳細資訊，請參閱將物件上傳至 S3 on Outposts 儲存貯體。

開始使用適用於 Java 的 AWS CLI 和 開發套件

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (0UTP0STS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

若要開始使用 S3 on Outposts，您必須建立儲存貯體、存取點和端點。接著，您可以將物件上傳至您的儲存貯體。下列範例說明如何使用適用於 Java 的 AWS CLI 和 SDK 開始使用 S3 on Outposts。若要開始使用主控台，請參閱 開始使用 AWS Management Console。

主題

- 步驟 1：建立儲存貯體
- 步驟 2：建立存取點
- 步驟 3：建立端點
- 步驟 4：將物件上傳至 S3 on Outposts 儲存貯體

步驟 1：建立儲存貯體

下列 AWS CLI 和適用於 Java 的 SDK 範例示範如何建立 S3 on Outposts 儲存貯體。

AWS CLI

Example

下列範例使用 AWS CLI建立 S3 on Outposts 儲存貯體 (`s3-outposts:CreateBucket`)。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-
id op-01ac5d28a6a232904
```

SDK for Java

Example

下列範例使用適用於 Java 的開發套件建立 S3 on Outposts 儲存貯體 (`s3-outposts:CreateBucket`)。

```
import com.amazonaws.services.s3control.model.*;

public String createBucket(String bucketName) {

    CreateBucketRequest reqCreateBucket = new CreateBucketRequest()
            .withBucket(bucketName)
            .withOutpostId(OutpostId)
            .withCreateBucketConfiguration(new CreateBucketConfiguration());

    CreateBucketResult respCreateBucket =
 s3ControlClient.createBucket(reqCreateBucket);
    System.out.printf("CreateBucket Response: %s%n", respCreateBucket.toString());

    return respCreateBucket.getBucketArn();

}
```

步驟 2：建立存取點

若要存取 Amazon S3 on Outposts 儲存貯體，您必須建立和設定存取點。這些範例說明如何使用 AWS CLI 和適用於 Java 的 開發套件來建立存取點。

存取點針對 Amazon S3 中的共用資料集，簡化管理大規模的資料存取。存取點為連接到儲存貯體的指定網路端點，您可以使用這些端點來執行 Amazon S3 物件操作，例如 `GetObject` 和 `PutObject`。使用 S3 on Outposts，您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。存取點僅支援虛擬主機樣式定址。

AWS CLI

Example

下列 AWS CLI 範例會建立 Outposts 儲存貯體的存取點。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control create-access-point --account-id 123456789012
 --name example-outposts-access-point --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

SDK for Java

Example

下列適用於 Java 的開發套件範例建立 Outposts 儲存貯體的存取點。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
import com.amazonaws.services.s3control.model.*;

public String createAccessPoint(String bucketArn, String accessPointName) {

    CreateAccessPointRequest reqCreateAP = new CreateAccessPointRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn)
            .withName(accessPointName)
            .withVpcConfiguration(new VpcConfiguration().withVpcId("vpc-12345"));

    CreateAccessPointResult respCreateAP =
 s3ControlClient.createAccessPoint(reqCreateAP);
    System.out.printf("CreateAccessPoint Response: %s%n", respCreateAP.toString());

    return respCreateAP.getAccessPointArn();

}
```

步驟 3：建立端點

若要將請求路由至 Amazon S3 on Outpost 存取點，您必須建立和設定 S3 on Outposts 端點。若要建立端點，您需要與 Outposts 主要區域的服務連結有效連接。Outpost 上的每個 Virtual Private Cloud (VPC) 可以有一個相關聯的端點。如需端點配額的詳細資訊，請參閱 S3 on Outposts 網路需求。您必須建立端點，才能存取您的 Outposts 儲存貯體並執行物件操作。如需詳細資訊，請參閱端點。

這些範例說明如何使用 AWS CLI 和適用於 Java 的 開發套件來建立端點。如需建立和管理端點所需許可的詳細資訊，請參閱 適用於 S3 on Outposts 端點的許可。

AWS CLI

Example

下列 AWS CLI 範例會使用 VPC 資源存取類型，為 Outpost 建立端點。VPC 衍生自子網路。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

下列 AWS CLI 範例使用客戶擁有的 IP 地址集區 (CoIP 集區） 存取類型，為 Outpost 建立端點。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --
customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

SDK for Java

Example

下列適用於 Java 的開發套件範例建立 Outposts 的端點。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.CreateEndpointRequest;
import com.amazonaws.services.s3outposts.model.CreateEndpointResult;

public void createEndpoint() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
```

```
                .standard().build();

    CreateEndpointRequest createEndpointRequest = new CreateEndpointRequest()
                .withOutpostId("op-0d79779cef3c30a40")
                .withSubnetId("subnet-8c7a57c5")
                .withSecurityGroupId("sg-ab19e0d1")
                .withAccessType("CustomerOwnedIp")
                .withCustomerOwnedIpv4Pool("ipv4pool-coip-12345678901234567");
    // Use .withAccessType and .withCustomerOwnedIpv4Pool only when the access type
 is
    // customer-owned IP address pool (CoIP pool)
    CreateEndpointResult createEndpointResult =
 s3OutpostsClient.createEndpoint(createEndpointRequest);
    System.out.println("Endpoint is created and its ARN is " +
 createEndpointResult.getEndpointArn());
}
```

步驟 4：將物件上傳至 S3 on Outposts 儲存貯體

若要上傳物件，請參閱 將物件上傳至 S3 on Outposts 儲存貯體。

適用於 S3 on Outposts 的網路

您可以使用 Amazon S3 on Outposts，針對需要本機資料存取、資料處理和資料駐留的應用程式，來存放和擷取內部部署物件。此部分說明存取 S3 on Outposts 的網路要求。

主題

選擇您的網路存取類型

您可以從 VPC 內或內部部署網路存取 S3 on Outposts。您可以使用存取點和透過端點連線，與您的 Outposts 儲存貯體進行通訊。此連線會在 AWS 網路內確保 VPC 與 S3 on Outposts 儲存貯體之間的流量。建立端點後，您必須將端點存取類型指定為 `Private` (適用於 VPC 路由) 或 `CustomerOwnedIp` (適用於客戶自訂 IP 地址集區 (CoIP 集區))。

- `Private` (針對 VPC 路由路由) - 若不指定存取類型，則預設 S3 on Outposts 會使用 `Private`。使用 `Private`存取類型，則 VPC 中的執行個體不需要公有 IP 地址，即可與 Outposts 中的資源通訊。您可以使用在 VPC 內使用 S3 on Outposts。這類端點無法透過直接 VPC 路由從內部部署網路存取。如需詳細資訊，請參閱《AWS Outposts 使用者指南》中的本機閘道路油表。
- `CustomerOwnedIp` (針對 CoIP 集區) - 如果您預設不使用 `Private` 存取類型，然後選擇 `CustomerOwnedIp`，您必須指定 IP 地址範圍。您可以使用此存取類型在內部部署網路和 VPC 內使用 S3 on Outposts。在 VPC 內存取 S3 on Outposts 時，您的流量將受限於本機閘道的頻寬。

存取 S3 on Outposts 儲存貯體和物件

若要存取 S3 on Outposts 儲存貯體和物件，必須具備下列項目：

- VPC 的存取點。
- 相同 VPC 的端點。
- Outpost 與 AWS 區域之間的作用中連線。如需如何將 Outpost 連線至區域的詳細資訊，請參閱 Outposts 使用者指南中的 Outpost 連線至 AWS 區域。 AWS

如需有關存取 S3 on Outposts 中儲存貯體和物件的詳細資訊，請參閱 使用 S3 on Outposts 儲存貯體 和 使用 S3 on Outposts 物件。

跨帳戶彈性網路界面

S3 on Outposts 端點是使用 Amazon 資源名稱 (ARN) 命名的資源。建立這些端點時， 會 AWS Outposts 設定多個跨帳戶彈性網路介面。S3 on Outposts 跨帳戶彈性網路界面與其他網路界面類型，但存在以下例外狀況：S3 on Outposts 會將跨帳戶彈性網路界面與 Amazon EC2 執行個體關聯。

S3 on Outposts 網域名稱系統 (DNS) 負載會透過跨帳戶彈性網路界面來平衡您的請求。S3 on Outposts 會在您的帳戶中建立跨帳戶彈性網路介面 AWS ，可從 Amazon EC2 主控台的網路介面窗格中看見。

對於使用 CoIP 集區存取類型的端點，S3 on Outposts 會從設定的 CoIP 集區配置 IP 地址，並將其與跨帳戶彈性網路界面建立關聯。

使用 S3 on Outposts 儲存貯體

使用 Amazon S3 on Outposts，您可以在您的 上建立 S3 儲存貯體 AWS Outposts ，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，輕鬆地在內部部署存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (OUTPOSTS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。就像在 Amazon S3 一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outposts 儲存貯體進行通訊。若要存取 S3 on Outposts 儲存貯體和物件，必須具備 VPC 的存取點，並且必須具備相同 VPC 的端點。如需詳細資訊，請參閱適用於 S3 on Outposts 的網路。

儲存貯體

在 S3 on Outposts 中，儲存貯體名稱對 Outpost 是唯一的，並且需要 AWS 區域 Outpost 所在區域的程式碼、 AWS 帳戶 ID、Outpost ID 和儲存貯體名稱來識別它們。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

如需詳細資訊，請參閱適用於 S3 on Outposts 的資源 ARN。

存取點

Amazon S3 on Outposts 支援僅限 虛擬私有雲端 (VPC) 的存取點來作為存取 Outpost 儲存貯體的唯一方法。

存取點針對 Amazon S3 中的共用資料集，簡化管理大規模的資料存取。存取點為連接到儲存貯體的指定網路端點，您可以使用這些端點來執行 Amazon S3 物件操作，例如 `GetObject` 和 `PutObject`。使用 S3 on Outposts，您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。存取點僅支援虛擬主機樣式定址。

下列範例顯示了 S3 on Outposts 存取點使用的 ARN 格式。存取點 ARN 包含 Outpost 所在區域的 AWS 區域 程式碼、 AWS 帳戶 ID、Outpost ID 和存取點名稱。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

端點

若要將請求路由至 S3 on Outpost 存取點，您必須建立和設定 S3 on Outposts 端點。對於 S3 on Outposts 端點，您可以將 VPC 私密地連線至 Outposts 儲存貯體。S3 on Outposts 端點是 Outposts 儲存貯體 S3 進入點的虛擬統一資源識別符 (URI)。這些端點是水平擴展、冗餘且高度可用的 VPC 元件。

Outpost 上的每個 Virtual Private Cloud (VPC) 可以有一個相關聯的端點，而且每個 Outpost 最多可以有 100 端點。您必須建立這些端點，才能存取您的 Outpost 儲存貯體並執行物件操作。建立這些端點也會讓相同的操作在 S3 和 S3 on Outposts 中運作，使 API 模型和行為相同。

適用於 S3 on Outposts 的 API

若要管理 Outpost 儲存貯體 API 操作，S3 on Outposts 會託管一個與 Amazon S3 端點不同的獨立端點。此端點為 `s3-outposts.region.amazonaws.com`。

若要使用 Amazon S3 API 操作，必須使用正確的 ARN 格式簽署儲存貯體和物件。您必須將 API 操作傳遞給 ARN，以便 Amazon S3 確定請求是針對Amazon S3 (`s3-control.region.amazonaws.com`) 還是 S3 on Outposts (`s3-outposts.region.amazonaws.com`)。根據 ARN 格式，S3 隨後可以適當地簽署和路由請求。

每當將請求傳送至 Amazon S3 控制平面時，開發套件會從 ARN 中擷取元件，並包含一個額外的標頭 `x-amz-outpost-id`，其中包含從 ARN 擷取的 *outpost-id* 值。來自 ARN 的服務名稱用來在路由傳送到 S3 on Outposts 端點之前簽署請求。該行為適用於所有由 `s3control` 用戶端處理的 API 操作。

下表列出了 Amazon S3 on Outposts 的擴展 API 操作，及其相對於 Amazon S3 的變更。

API	S3 on Outposts 參數值	
`CreateBucket`	作為 ARN 的儲存貯體名稱，Outpost ID	
`ListRegionalBuckets`	Outpost ID	
`DeleteBucket`	作為 ARN 的儲存貯體名稱	
`DeleteBucketLifecycleConfiguration`	作為 ARN 的儲存貯體名稱	

API	S3 on Outposts 參數值	
GetBucketLifecycle Configuration	作為 ARN 的儲存貯體名稱	
PutBucketLifecycle Configuration	作為 ARN 的儲存貯體名稱	
GetBucketPolicy	作為 ARN 的儲存貯體名稱	
PutBucketPolicy	作為 ARN 的儲存貯體名稱	
DeleteBucketPolicy	作為 ARN 的儲存貯體名稱	
GetBucketTagging	作為 ARN 的儲存貯體名稱	
PutBucketTagging	作為 ARN 的儲存貯體名稱	
DeleteBucketTagging	作為 ARN 的儲存貯體名稱	
CreateAccessPoint	作為 ARN 的存取點名稱	
DeleteAccessPoint	作為 ARN 的存取點名稱	
GetAccessPoint	作為 ARN 的存取點名稱	
GetAccessPoint	作為 ARN 的存取點名稱	
ListAccessPoints	作為 ARN 的存取點名稱	
PutAccessPointPolicy	作為 ARN 的存取點名稱	
GetAccessPointPolicy	作為 ARN 的存取點名稱	
DeleteAccessPointPolicy	作為 ARN 的存取點名稱	

建立和管理 S3 on Outposts 儲存貯體

如需有關建立和管理 S3 on Outposts 儲存貯體的詳細資訊，請參閱下列主題。

建立 S3 on Outposts 儲存貯體

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (0UTP0STS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

> **Note**
>
> 建立儲存貯體 AWS 帳戶 的 擁有該儲存貯體，而且是唯一可以對其遞交動作的儲存貯體。儲存貯體具有組態屬性，例如 Outpost、標籤、預設加密和存取點設定。存取點設定包含用於存取儲存貯體中物件的虛擬私有雲端 (VPC) 和存取點政策，以及其他中繼資料。如需詳細資訊，請參閱S3 on Outposts 規格。
>
> 如果您想要建立使用 AWS PrivateLink 的儲存貯體，透過虛擬私有雲端 (VPC) 中的介面 VPC 端點提供儲存貯體和端點管理存取權，請參閱 AWS PrivateLink 以取得 S3 on Outposts。

下列範例說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 建立 S3 on Outposts 儲存貯體 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台：//https：/https://console.aws.amazon.com/s3/.microsoft.com。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇 Create Outposts bucket (建立 Outposts 儲存貯體)。
4. 在 Bucket name (儲存貯體名稱) 中，為儲存貯體輸入符合網域名稱系統 (DNS) 規範的名稱。

   儲存貯體名稱必須；

   - 在 AWS 帳戶、Outpost 和 Outpost 所在的 中是唯一 AWS 區域 的。
   - 長度必須介於 3–63 個字元之間。
   - 不含大寫字元。

- 以小寫字母或數字開頭。

  建立儲存貯體後，便無法變更其名稱。如需為儲存貯體命名的相關資訊，請參閱《Amazon S3 使用者指南》中的一般用途儲存貯體命名規則。

  > **Important**
  >
  > 避免在儲存貯體名稱中包含敏感資訊，例如帳戶號碼。在指向儲存貯體中之物件的 URL 中，會顯示儲存貯體名稱。

5. 在 Outpost 中，選擇您要儲存貯體駐留的 Outpost。
6. 在 Bucket Versioning (儲存貯體版本控制) 下，將 S3 on Outposts 儲存貯體的 S3 版本控制狀態設定為下列其中一個選項：

   - Disable (停用) (預設) - 儲存貯體會保留未版本控制的狀態。
   - Enable (啟用) - 針對儲存貯體中的物件啟用 S3 版本控制。所有新增至儲存貯體的物件都會收到唯一的版本 ID。

   如需 S3 版本控制的詳細資訊，請參閱「針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制」。
7. (選擇性) 新增要與 Outposts 儲存貯體建立關聯的任何 optional tags (選用標籤)。您可以使用標籤來追蹤個別專案或一組專案的條件，或者使用成本分配標籤來標示儲存貯體。

   存放在 Outposts 儲存貯體中的所有物件預設為使用伺服器端加密與 Amazon S3 受管加密金鑰 (SSE-S3) 進行儲存。您也可以明確選擇使用伺服器端加密與客戶提供的加密金鑰 (SSE-C) 來存放物件。若要變更加密類型，您必須使用 REST API、 AWS Command Line Interface (AWS CLI) 或 AWS SDKs。
8. 在 Outposts access point settings (Outposts 存取點設定) 區段中，輸入存取點名稱。

   S3 on Outposts 存取點針對 S3 on Outposts 中的共用資料集，簡化了對大規模資料存取的管理。存取點為連接到 Outposts 儲存貯體的具名網路端點，您可以使用這些端點來執行 S3 物件操作。如需詳細資訊，請參閱存取點。

   存取點名稱在此區域和 Outpost 的帳戶中必須是唯一的，並且符合存取點約束與限制。
9. 對此 Amazon S3 on Outposts 存取點選擇 VPC。

   如果您沒有 VPC，請選擇 Create VPC (建立 VPC)。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的建立僅限虛擬私有雲端 (VPC) 使用的存取點。

Virtual Private Cloud (VPC) 可讓您將 AWS 資源啟動到您定義的虛擬網路。這個虛擬網路與您在資料中心中操作的傳統網路非常相似，且具備使用 可擴展基礎設施的優勢 AWS

10. (對於現有 VPC 可選) 請為您的端點選擇Endpoint subnet (端點子網路)。

    子網路是您的 VPC 中的 IP 地址範圍。如果您沒有想要的子網路，請選擇 Create subnet (建立子網路)。如需詳細資訊，請參閱適用於 S3 on Outposts 的網路。

11. (對於現有 VPC 可選) 請為您的端點選擇Endpoint security group (端點安全群組)。

    安全群組做為虛擬防火牆以控制傳入及傳出流量。

12. (對於現有 VPC 可選) 請選擇 Endpoint access type (端點存取類型)：

    - 私有 - 與 VPC 一起使用。
    - 客戶擁有的 IP - 與內部部署網路內的客戶擁有的 IP 地址集區 (CoIP 集區) 一起使用。

13. (選用) 指定Outpost 存取點政策。主控台會自動顯示存取點的 Amazon Resource Name (ARN)，您可以在政策中使用該名稱。

14. 選擇 Create Outposts bucket (建立 Outposts 儲存貯體)。

> **Note**
>
> 建立您的 Outpost 端點以及您的儲存貯體準備就緒可以使用，可能需要 5 分鐘。若要設定其他儲存貯體設定，請選擇 View details (檢視詳細資訊)

使用 AWS CLI

Example

下列範例使用 AWS CLI建立 S3 on Outposts 儲存貯體 (`s3-outposts:CreateBucket`)。若要執行此命令，請以您自己的資訊取代 *`user input placeholders`*。

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-
id op-01ac5d28a6a232904
```

使用適用於 Java 的 AWS 開發套件

Example

下列範例使用適用於 Java 的開發套件建立 S3 on Outposts 儲存貯體 (`s3-outposts:CreateBucket`)。

```
import com.amazonaws.services.s3control.model.*;

public String createBucket(String bucketName) {

    CreateBucketRequest reqCreateBucket = new CreateBucketRequest()
            .withBucket(bucketName)
            .withOutpostId(OutpostId)
            .withCreateBucketConfiguration(new CreateBucketConfiguration());

    CreateBucketResult respCreateBucket =
 s3ControlClient.createBucket(reqCreateBucket);
    System.out.printf("CreateBucket Response: %s%n", respCreateBucket.toString());

    return respCreateBucket.getBucketArn();

}
```

新增 S3 on Outposts 儲存貯體的標籤

您可以新增 S3 on Outposts 儲存貯體標籤，以追蹤個別專案或專案群組的儲存成本和其他條件。

> **Note**
>
> 建立儲存貯體 AWS 帳戶 的 擁有它，是唯一可以變更其標籤的 。

使用 S3 主控台

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台開啟 https：//https://console.aws.amazon.com/s3/。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇包含您要編輯其標籤的 Outposts 儲存貯體。
4. 選擇屬性索引標籤。

5. 在 Tags (標籤) 下方，選擇 Edit (編輯)。
6. 選擇 Add new tag (新增標籤)，然後輸入 金鑰和選項值。

   新增要與 Outposts 儲存貯體關聯的任何標籤，以追蹤個別專案或專案群組的條件。
7. 選擇 Save changes (儲存變更)。

使用 AWS CLI

下列 AWS CLI 範例使用指定標籤 () 的目前資料夾中的 JSON 文件，將標記組態套用至 S3 on Outposts 儲存貯體*tagging.json*。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --tagging file://tagging.json

tagging.json

{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
}
```

下列 AWS CLI 範例會直接從命令列將標記組態套用至 S3 on Outposts 儲存貯體。

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --tagging 'TagSet=[{Key=organization,Value=marketing}]'
```

如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 put-bucket-tagging。

使用儲存貯體政策管理 Amazon S3 on Outposts 儲存貯體的存取

儲存貯體政策是資源型 AWS Identity and Access Management (IAM) 政策，可用來將存取許可授予儲存貯體及其中的物件。只有儲存貯體擁有者可建立政策與儲存貯體的關聯。連接到儲存貯體的許可會

套用至儲存貯體擁有者帳戶擁有的所有儲存貯體物件。儲存貯體政策的大小限制為 20 KB。如需詳細資訊，請參閱儲存貯體政策。

您可以更新儲存貯體政策以管理對 Amazon S3 on Outposts 儲存貯體的存取。如需詳細資訊，請參閱下列主題。

主題

- 新增或編輯 Amazon S3 on Outposts 儲存貯體的儲存貯體政策
- 檢視 Amazon S3 on Outposts 儲存貯體的儲存貯體政策
- 刪除 Amazon S3 on Outposts 儲存貯體的儲存貯體政策
- 儲存貯體政策範例

新增或編輯 Amazon S3 on Outposts 儲存貯體的儲存貯體政策

儲存貯體政策是資源型 AWS Identity and Access Management (IAM) 政策，可用來將存取許可授予儲存貯體及其中的物件。只有儲存貯體擁有者可建立政策與儲存貯體的關聯。連接到儲存貯體的許可會套用至儲存貯體擁有者帳戶擁有的所有儲存貯體物件。儲存貯體政策的大小限制為 20 KB。如需詳細資訊，請參閱儲存貯體政策。

下列主題說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 來更新 Amazon S3 on Outposts 儲存貯體政策 適用於 Java 的 AWS SDK。

使用 S3 主控台

建立或編輯儲存貯體政策

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台開啟 https://console.aws.amazon.com/s3/ S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要編輯其儲存貯體政策的 Outposts 儲存貯體。
4. 選擇許可索引標籤標籤。
5. 在 Outposts 儲存貯體政策部分，若要建立或編輯新政策，請選擇Edit (編輯)。

   現在您可以新增或編輯 S3 on Outposts 儲存貯體政策。如需詳細資訊，請參閱使用 S3 on Outposts 設定 IAM。

使用 AWS CLI

下列 AWS CLI 範例會在 Outposts 儲存貯體上放置政策。

1. 將以下儲存貯體政策儲存到 JSON 檔案中。在此範例中，檔案命名為 policy1.json。以您自己的資訊取代 *user input placeholders*。

```
{
   "Version":"2012-10-17",
   "Id":"testBucketPolicy",
   "Statement":[
      {
         "Sid":"st1",
         "Effect":"Allow",
         "Principal":{
            "AWS":"123456789012"
         },
         "Action":"s3-outposts:*",
         "Resource":"arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket"
      }
   ]
}
```

2. 提交 JSON 檔案以做為 put-bucket-policy CLI 命令的一部分。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control put-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --policy file://policy1.json
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例在 Outposts 儲存貯體上放置政策。

```
import com.amazonaws.services.s3control.model.*;

public void putBucketPolicy(String bucketArn) {
```

```
    String policy = "{\"Version\":\"2012-10-17\",\"Id\":\"testBucketPolicy\",
\"Statement\":[{\"Sid\":\"st1\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"" +
 AccountId+ "\"},\"Action\":\"s3-outposts:*\",\"Resource\":\"" + bucketArn + "\"}]}";

    PutBucketPolicyRequest reqPutBucketPolicy = new PutBucketPolicyRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn)
            .withPolicy(policy);

    PutBucketPolicyResult respPutBucketPolicy =
 s3ControlClient.putBucketPolicy(reqPutBucketPolicy);
    System.out.printf("PutBucketPolicy Response: %s%n",
 respPutBucketPolicy.toString());

}
```

檢視 Amazon S3 on Outposts 儲存貯體的儲存貯體政策

儲存貯體政策是資源型 AWS Identity and Access Management (IAM) 政策，可用來將存取許可授予儲存貯體及其中的物件。只有儲存貯體擁有者可建立政策與儲存貯體的關聯。連接到儲存貯體的許可會套用至儲存貯體擁有者帳戶擁有的所有儲存貯體物件。儲存貯體政策的大小限制為 20 KB。如需詳細資訊，請參閱儲存貯體政策。

下列主題說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 來檢視 Amazon S3 on Outposts 儲存貯體政策 適用於 Java 的 AWS SDK。

使用 S3 主控台

建立或編輯儲存貯體政策

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台開啟 https://console.aws.amazon.com/s3/ S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要編輯其許可的 Outposts 儲存貯體。
4. 選擇 Permissions (許可) 標籤。
5. 在 Outposts 儲存貯體政策中，您可以檢閱現有的儲存貯體政策。如需詳細資訊，請參閱使用 S3 on Outposts 設定 IAM。

使用 AWS CLI

下列 AWS CLI 範例取得 Outposts 儲存貯體的政策。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control get-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例取得 Outposts 儲存貯體的政策。

```
import com.amazonaws.services.s3control.model.*;

public void getBucketPolicy(String bucketArn) {

    GetBucketPolicyRequest reqGetBucketPolicy = new GetBucketPolicyRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn);

    GetBucketPolicyResult respGetBucketPolicy =
 s3ControlClient.getBucketPolicy(reqGetBucketPolicy);
    System.out.printf("GetBucketPolicy Response: %s%n",
 respGetBucketPolicy.toString());

}
```

刪除 Amazon S3 on Outposts 儲存貯體的儲存貯體政策

儲存貯體政策是資源型 AWS Identity and Access Management (IAM) 政策，可用來將存取許可授予儲存貯體及其中的物件。只有儲存貯體擁有者可建立政策與儲存貯體的關聯。連接到儲存貯體的許可會套用至儲存貯體擁有者帳戶擁有的所有儲存貯體物件。儲存貯體政策的大小限制為 20 KB。如需詳細資訊，請參閱儲存貯體政策。

下列主題說明如何使用 AWS Management Console 或 AWS Command Line Interface () 檢視 Amazon S3 on Outposts 儲存貯體政策AWS CLI。

使用 S3 主控台

若要刪除儲存貯體政策

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要編輯其許可的 Outposts 儲存貯體。
4. 選擇 Permissions (許可) 標籤。
5. 在 Outposts bucket policy (Outposts 儲存貯體政策) 區段中，選擇 Delete (刪除)。
6. 確認刪除。

使用 AWS CLI

下列範例示範使用 AWS CLI刪除 S3 on Outposts 儲存貯體 (`s3-outposts:DeleteBucket`) 的儲存貯體政策。若要執行此命令，請以您自己的資訊取代 *`user input placeholders`*。

```
aws s3control delete-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

儲存貯體政策範例

使用 S3 on Outposts 儲存貯體政策，您可以安全地存取 S3 on Outposts 儲存貯體中的物件，使得只有具有適當許可的使用者才能存取它們。您甚至可以防止已驗證但沒有適當許可的使用者存取 S3 on Outposts 資源。

本節顯示 S3 on Outposts 儲存貯體政策之一般使用案例的範例。若要測試這些政策，請使用您自己的資訊 (例如儲存貯體名稱) 取代 *`user input placeholders`*。

若要授予或拒絕一組物件的許可，您可以在 Amazon Resource Name (ARN) 和其他值上使用萬用字元 (*)。例如，您可以控制對以常用字首開頭或以給定的副檔名結束的一組物件存取權，例如 `.html`。

如需 AWS Identity and Access Management (IAM) 政策語言的詳細資訊，請參閱 使用 S3 on Outposts 設定 IAM。

> **Note**
>
> 使用 Amazon S3 主控台來測試 s3outposts 許可時，您必須授予主控台所需的其他許可：`s3outposts:createendpoint`、`s3outposts:listendpoints`，以此類推。

用於建立儲存貯體政策的其他資源

- 如需建立 S3 on Outposts 儲存貯體政策時可使用的 IAM 政策動作、資源和條件金鑰清單，請參閱 Amazon S3 on Outposts 的動作、資源和條件金鑰。
- 如需建立 S3 on Outposts 政策的指引，請參閱新增或編輯 Amazon S3 on Outposts 儲存貯體的儲存貯體政策。

主題

- 根據特定 IP 位址管理對 Amazon S3 on Outposts 儲存貯體的存取

根據特定 IP 位址管理對 Amazon S3 on Outposts 儲存貯體的存取

儲存貯體政策是資源型 AWS Identity and Access Management (IAM) 政策，可用來將存取許可授予儲存貯體及其中的物件。只有儲存貯體擁有者可建立政策與儲存貯體的關聯。連接到儲存貯體的許可會套用至儲存貯體擁有者帳戶擁有的所有儲存貯體物件。儲存貯體政策的大小限制為 20 KB。如需詳細資訊，請參閱儲存貯體政策。

限制針對特定 IP 位址的存取

下列範例拒絕所有使用者對指定儲存貯體中的物件執行任何 S3 on Outposts 操作，除非請求源自指定的 IP 位址範圍。

> **Note**
>
> 限制對特定 IP 位址的存取時，請確定您也指定了哪些 VPC 端點、VPC 來源 IP 位址或外部 IP 位址可以存取 S3 on Outposts 儲存貯體。否則，如果您的政策拒絕所有使用者對您 S3 on Outposts 儲存貯體中的物件執行任何 s3outposts 操作，而且未設定任何適當的許可，則可能會失去對儲存貯體的存取權。

此政策的 `Condition` 陳述式會將 *`192.0.2.0/24`* 識別為允許之網際網路通訊協定第 4 版 (IPv4) IP 位址的範圍。

Condition 區塊使用 NotIpAddress條件和 aws:SourceIp 條件索引鍵，這是 AWS 寬條件索引鍵。aws:SourceIp 條件金鑰僅可用於公有 IP 位址範圍。如需這些條件金鑰詳細資訊，請參閱適用於 S3 on Outposts 的動作、資源和條件金鑰。aws:SourceIp IPv4 值會使用標準 CIDR 表示法。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

> **Warning**
>
> 使用此 S3 on Outposts 政策之前，請以適當的使用案例值取代此範例中的 *192.0.2.0/24* IP 位址範圍。否則，您將失去存取儲存貯體的能力。

```
{
    "Version": "2012-10-17",
    "Id": "S3OutpostsPolicyId1",
    "Statement": [
        {
            "Sid": "IPAllow",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3-outposts:*",
            "Resource": [
                "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
accesspoint/EXAMPLE-ACCESS-POINT-NAME",
                "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/amzn-s3-demo-bucket"
            ],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": "192.0.2.0/24"
                }
            }
        }
    ]
}
```

同時允許 IPv4 和 IPv6 位址

當您開始使用 IPv6 位址時，建議除了現有 IPv4 範圍之外，還使用 IPv6 位址範圍來更新組織的所有政策。這樣做有助於確保政策在轉換為 IPv6 時繼續運作。

下列 S3 on Outposts 儲存貯體政策範例會示範如何混合使用 IPv4 與 IPv6 位址範圍，以涵蓋組織中所有的有效 IP 位址。政策範例允許存取 IP 位址範例 *192.0.2.1* 與 *2001:DB8:1234:5678::1*，並且拒絕存取位址 *203.0.113.1* 與 *2001:DB8:1234:5678:ABCD::1*。

aws:SourceIp 條件金鑰僅可用於公有 IP 位址範圍。aws:SourceIp 的 IPv6 值必須為標準 CIDR 格式。針對 IPv6，我們支援使用 :: 代表 0 的範圍 (例如，2001:DB8:1234:5678::/64)。如需詳細資訊，請參閱《IAM 使用者指南》中的 IP 位址條件運算子。

> **Warning**
>
> 使用此 S3 on Outposts 政策之前，請以適當的使用案例值取代此範例中的 IP 位址範圍。否則，您可能會失去存取儲存貯體的能力。

```
{
    "Id": "S3OutpostsPolicyId2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIPmix",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3outposts:*",
            "Resource": [
                "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/amzn-s3-demo-bucket",
                        "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-
ID/bucket/amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "2001:DB8:1234:5678::/64"
                    ]
                },
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "203.0.113.0/24",
                        "2001:DB8:1234:5678:ABCD::/80"
                    ]
                }
```

```
            }
        }
    ]
}
```

列出 Amazon S3 on Outposts 儲存貯體

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (0UTP0STS)，其使用 Amazon S3 APIs，旨在在您的多個裝置和伺服器上以持久且備援的方式存放資料 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、SDK 或 REST API 使用 S3 on Outposts。 AWS SDKs 如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

如需使用 S3 on Outposts 儲存貯體的詳細資訊，請參閱 使用 S3 on Outposts 儲存貯體。

下列範例說明如何使用 AWS Management Console AWS CLI、 和 傳回 S3 on Outposts 儲存貯體的清單 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 在 Outposts 儲存貯體下，檢視 S3 on Outposts 儲存貯體清單。

使用 AWS CLI

下列 AWS CLI 範例會取得 Outpost 中的儲存貯體清單。若要執行此命令，請以您自己的資訊取代每個 *`user input placeholder`*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 list-regional-buckets。

```
aws s3control list-regional-buckets --account-id 123456789012 --outpost-
id op-01ac5d28a6a232904
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例取得 Outpost 中的儲存貯體清單。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 ListRegionalBuckets。

```
import com.amazonaws.services.s3control.model.*;

public void listRegionalBuckets() {

    ListRegionalBucketsRequest reqListBuckets = new ListRegionalBucketsRequest()
            .withAccountId(AccountId)
            .withOutpostId(OutpostId);

    ListRegionalBucketsResult respListBuckets =
 s3ControlClient.listRegionalBuckets(reqListBuckets);
    System.out.printf("ListRegionalBuckets Response: %s%n",
 respListBuckets.toString());

}
```

使用適用於 Java 的 AWS CLI 和 開發套件取得 S3 on Outposts 儲存貯體

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (0UTP0STS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

下列範例說明如何使用 AWS CLI 和 取得 S3 on Outposts 儲存貯體 適用於 Java 的 AWS SDK。

> **Note**
>
> 當您透過 AWS CLI AWS SDKs 使用 Amazon S3 on Outposts 時，您會提供 Outpost 的存取點 ARN 來取代儲存貯體名稱。存取點 ARN 採用以下形式，其中 *region* 是 Outpost 所在區域的 AWS 區域 代碼：

```
arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
accesspoint/example-outposts-access-point
```

如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

使用 AWS CLI

下列 S3 on Outposts 範例使用 AWS CLI取得儲存貯體。若要執行此命令，請以您自己的資訊取代每個 *user input placeholder*。如需此命令詳細資訊，請參閱 AWS CLI 參考中的 get-bucket。

```
aws s3control get-bucket --account-id 123456789012 --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket"
```

使用適用於 Java 的 AWS 開發套件

下列 S3 on Outposts 範例使用適用於 Java 的開發套件，取得一個儲存貯體。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 GetBucket。

```
import com.amazonaws.services.s3control.model.*;

public void getBucket(String bucketArn) {

    GetBucketRequest reqGetBucket = new GetBucketRequest()
            .withBucket(bucketArn)
            .withAccountId(AccountId);

    GetBucketResult respGetBucket = s3ControlClient.getBucket(reqGetBucket);
    System.out.printf("GetBucket Response: %s%n", respGetBucket.toString());

}
```

刪除 Amazon S3 on Outposts 儲存貯體

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (0UTP0STS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在

Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

如需使用 S3 on Outposts 儲存貯體的詳細資訊，請參閱 使用 S3 on Outposts 儲存貯體。

建立儲存貯體 AWS 帳戶 的 擁有該儲存貯體，而且是唯一可以將其刪除的儲存貯體。

> **Note**
>
> - Outposts 儲存貯體在刪除之前必須先清空。
>
> Amazon S3 主控台不支援 S3 on Outposts 物件動作。若要刪除 S3 on Outposts 儲存貯體中的物件，您必須使用 REST API AWS CLI或 AWS SDKs。
> - 您必須先刪除儲存貯體的所有 Outposts 存取點，才能刪除 Outposts 儲存貯體。如需詳細資訊，請參閱刪除存取點。
> - 您無法恢復刪除後的儲存貯體。

下列範例示範如何使用 AWS Management Console 和 AWS Command Line Interface () 刪除 S3 on Outposts 儲存貯體AWS CLI。

使用 S3 主控台

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台開啟 https://console.aws.amazon.com/s3/ S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要刪除的儲存貯體，然後選擇 Delete (刪除)。
4. 確認刪除。

使用 AWS CLI

下列範例示範使用 AWS CLI刪除 S3 on Outposts 儲存貯體 (`s3-outposts:DeleteBucket`)。若要執行此命令，請以您自己的資訊取代 *`user input placeholders`*。

```
aws s3control delete-bucket --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

使用 Amazon S3 on Outposts 存取點

若要存取 Amazon S3 on Outposts 儲存貯體，您必須建立和設定存取點。

存取點針對 Amazon S3 中的共用資料集，簡化管理大規模的資料存取。存取點為連接到儲存貯體的指定網路端點，您可以使用這些端點來執行 Amazon S3 物件操作，例如 `GetObject` 和 `PutObject`。使用 S3 on Outposts，您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。存取點僅支援虛擬主機樣式定址。

> **Note**
>
> AWS 帳戶 建立 Outposts 儲存貯體的 擁有它，是唯一可以為其指派存取點的 。

下列各節說明了如何建立和管理 S3 on Outposts 儲存貯體存取點。

主題

建立 S3 on Outposts 存取點

若要存取 Amazon S3 on Outposts 儲存貯體，您必須建立和設定存取點。

存取點針對 Amazon S3 中的共用資料集，簡化管理大規模的資料存取。存取點為連接到儲存貯體的指定網路端點，您可以使用這些端點來執行 Amazon S3 物件操作，例如 `GetObject` 和 `PutObject`。使用 S3 on Outposts，您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。存取點僅支援虛擬主機樣式定址。

下列範例說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 建立 S3 on Outposts 存取點 適用於 Java 的 AWS SDK。

> **Note**
>
> AWS 帳戶 建立 Outposts 儲存貯體的 擁有它，是唯一可以為其指派存取點的 。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要為其建立 Outposts 存取點的 Outposts 儲存貯體。
4. 選擇 Outposts access points (Outposts 存取點) 索引標籤。
5. 在 Outposts access points (Outposts 存取點) 區段中，選擇 Create Outposts access point (建立 Outposts 存取點)。
6. 在 Outposts access point settings (Outposts 存取點設定) 中，輸入存取點的名稱，然後選擇存取點的 Virtual Private Cloud (VPC)。
7. 如果您想要新增存取點的政策，可以在 Outposts access point policy (Outposts 存取點政策) 區段輸入政策。

   如需詳細資訊，請參閱使用 S3 on Outposts 設定 IAM。

使用 AWS CLI

Example

下列 AWS CLI 範例會建立 Outposts 儲存貯體的存取點。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control create-access-point --account-id 123456789012
 --name example-outposts-access-point --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

使用適用於 Java 的 AWS 開發套件

Example

下列適用於 Java 的開發套件範例建立 Outposts 儲存貯體的存取點。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
import com.amazonaws.services.s3control.model.*;

public String createAccessPoint(String bucketArn, String accessPointName) {

    CreateAccessPointRequest reqCreateAP = new CreateAccessPointRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn)
            .withName(accessPointName)
            .withVpcConfiguration(new VpcConfiguration().withVpcId("vpc-12345"));

    CreateAccessPointResult respCreateAP =
 s3ControlClient.createAccessPoint(reqCreateAP);
    System.out.printf("CreateAccessPoint Response: %s%n", respCreateAP.toString());

    return respCreateAP.getAccessPointArn();

}
```

針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名

使用 S3 on Outposts，您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。每次建立儲存貯體的存取點時，S3 on Outposts 都會自動產生存取點別名。您可以針對任何資料平面操作使用此存取點別名，而不是存取點 Amazon Resource Name (ARN)。例如，您可以使用存取點別名來執行物件層級操作，例如 PUT、GET、LIST 等等。如需這些操作的清單，請參閱 適用於管理物件的 Amazon S3 API 操作。

下列範例顯示了名稱為 *my-access-point* 之存取點的 ARN 和存取點別名。

- 存取點 ARN – `arn:aws:s3-outposts:`*region*`:`*123456789012*`:outpost/`*op-01ac5d28a6a232904*`/accesspoint/`*my-access-point*
- 存取點別名 – *my-access-po*-*o01ac5d28a6a232904e8xz5w8ijx1qzlbp3i3kuse10*--op-s3

如需 ARN 的詳細資訊，請參閱《AWS 一般參考》中的 Amazon Resource Name (ARN)。

如需存取點別名的詳細資訊，請參閱下列主題。

主題

- 存取點別名

- 在 S3 on Outposts 物件操作中使用存取點別名
- 限制

存取點別名

存取點別名是在與 S3 on Outposts 儲存貯體相同的命名空間內建立的。當您建立存取點時，S3 on Outposts 會自動產生無法變更的存取點別名。存取點別名符合有效 S3 on Outposts 儲存貯體名稱的所有要求，並由下列部分組成：

access point name prefix-*metadata*--op-s3

> **Note**
>
> --op-s3 尾碼保留給存取點別名，因此建議不要將其用於儲存貯體或存取點名稱。如需 S3 on Outposts 儲存貯體命名規則的詳細資訊，請參閱 使用 S3 on Outposts 儲存貯體。

尋找存取點別名

下列範例向您展示如何使用 Amazon S3 主控台和 AWS CLI尋找存取點別名。

Example ：在 Amazon S3 主控台中尋找並複製存取點別名

在主控台中建立存取點之後，您可以從 Access Points (存取點) 清單中的 Access Point alias (存取點別名) 欄取得存取點別名。

複製存取點別名

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts access points (Outposts 存取點)。
3. 若要複製存取點別名，請執行下列其中一項動作：
   - 在 Access Points (存取點) 清單中，選取存取點名稱旁邊的選項按鈕，然後選擇 Copy Access Point alias (複製存取點別名)。
   - 選擇存取點名稱。然後，在 Outposts access point overview (Outposts 存取點概觀) 下，複製存取點別名。

Example ：使用 建立存取點， AWS CLI 並在回應中尋找存取點別名

下列 `create-access-point`命令 AWS CLI 範例會建立存取點，並傳回自動產生的存取點別名。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control create-access-point --bucket example-outposts-bucket --name example-
outposts-access-point --account-id 123456789012

{
    "AccessPointArn":
    "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
accesspoint/example-outposts-access-point",
    "Alias": "example-outp-o01ac5d28a6a232904e8xz5w8ijx1qzlbp3i3kuse10--op-s3"
}
```

Example ：使用 取得存取點別名 AWS CLI

下列 `get-access-point`命令 AWS CLI 範例會傳回指定存取點的相關資訊。此資訊包括存取點別名。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control get-access-point --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --name example-outposts-access-point --account-id 123456789012

{
    "Name": "example-outposts-access-point",
    "Bucket": "example-outposts-bucket",
    "NetworkOrigin": "Vpc",
    "VpcConfiguration": {
        "VpcId": "vpc-01234567890abcdef"
    },
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2022-09-18T17:49:15.584000+00:00",
    "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlbp3i3kuse10--op-s3"
}
```

Example ：使用 列出存取點以尋找存取點別名 AWS CLI

`list-access-points` 命令的下列 AWS CLI 範例會列出指定存取點的相關資訊。此資訊包括存取點別名。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket

{
    "AccessPointList": [
        {
            "Name": "example-outposts-access-point",
            "NetworkOrigin": "Vpc",
            "VpcConfiguration": {
                "VpcId": "vpc-01234567890abcdef"
            },
            "Bucket": "example-outposts-bucket",
            "AccessPointArn": "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point",
            "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlbp3i3kuse10--op-s3"
        }
    ]
}
```

在 S3 on Outposts 物件操作中使用存取點別名

採用存取點時，您可以使用存取點別名，而不需要進行大量的程式碼變更。

此 AWS CLI 範例顯示 S3 on Outposts 儲存貯體`get-object`的操作。此範例會使用存取點別名作為 `--bucket` 的值，而非完整存取點 ARN。

```
aws s3api get-object --bucket my-access-po-o0b1d075431d83bebde8xz5w8ijx1qzlbp3i3kuse10--op-s3 --key  testkey  sample-object.rtf

{
    "AcceptRanges": "bytes",
    "LastModified": "2020-01-08T22:16:28+00:00",
    "ContentLength": 910,
    "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
    "VersionId": "null",
    "ContentType": "text/rtf",
```

```
    "Metadata": {}
}
```

限制

- 客戶無法設定別名。
- 存取點上的別名無法刪除、修改或停用。
- 您無法將存取點別名用於 S3 on Outposts 控制平面操作。如需 S3 on Outposts 控制平面操作清單，請參閱適用於管理儲存貯體的 Amazon S3 Control API 操作。
- 別名不能用於 AWS Identity and Access Management (IAM) 政策。

檢視存取點組態的相關資訊

存取點針對 Amazon S3 中的共用資料集，簡化管理大規模的資料存取。存取點為連接到儲存貯體的指定網路端點，您可以使用這些端點來執行 Amazon S3 物件操作，例如 `GetObject` 和 `PutObject`。使用 S3 on Outposts，您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。存取點僅支援虛擬主機樣式定址。

下列主題說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 傳回 S3 on Outposts 存取點的組態資訊 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts access points (Outposts 存取點)。
3. 選擇您要檢視組態詳細資訊的 Outposts 存取點。
4. 在 Outposts 存取點概觀下，檢閱存取點的組態詳細資訊。

使用 AWS CLI

下列 AWS CLI 範例會取得 Outposts 儲存貯體的存取點。以您自己的資訊取代 *user input placeholders*。

```
aws s3control get-access-point --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例取得 Outposts 儲存貯體的存取點。

```
import com.amazonaws.services.s3control.model.*;

public void getAccessPoint(String accessPointArn) {

    GetAccessPointRequest reqGetAP = new GetAccessPointRequest()
            .withAccountId(AccountId)
            .withName(accessPointArn);

    GetAccessPointResult respGetAP = s3ControlClient.getAccessPoint(reqGetAP);
    System.out.printf("GetAccessPoint Response: %s%n", respGetAP.toString());

}
```

檢視 Amazon S3 Outposts 存取點清單

存取點針對 Amazon S3 中的共用資料集，簡化管理大規模的資料存取。存取點為連接到儲存貯體的指定網路端點，您可以使用這些端點來執行 Amazon S3 物件操作，例如 `GetObject` 和 `PutObject`。使用 S3 on Outposts，您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。存取點僅支援虛擬主機樣式定址。

下列主題說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 傳回 S3 on Outposts 存取點的清單 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts access points (Outposts 存取點)。
3. 在 Outposts 存取點下，檢閱 S3 on Outposts 存取點。

使用 AWS CLI

下列 AWS CLI 範例列出 Outposts 儲存貯體的存取點。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例列出了 Outposts 儲存貯體的存取點。

```
import com.amazonaws.services.s3control.model.*;

public void listAccessPoints(String bucketArn) {

    ListAccessPointsRequest reqListAPs = new ListAccessPointsRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn);

    ListAccessPointsResult respListAPs = s3ControlClient.listAccessPoints(reqListAPs);
    System.out.printf("ListAccessPoints Response: %s%n", respListAPs.toString());

}
```

刪除存取點

存取點針對 Amazon S3 中的共用資料集，簡化管理大規模的資料存取。存取點為連接到儲存貯體的指定網路端點，您可以使用這些端點來執行 Amazon S3 物件操作，例如 `GetObject` 和 `PutObject`。使用 S3 on Outposts，您必須使用存取點來存取 Outposts 儲存貯體中的任何物件。存取點僅支援虛擬主機樣式定址。

下列範例說明如何使用 AWS Management Console 和 AWS Command Line Interface () 刪除存取點 AWS CLI。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts access points (Outposts 存取點)。
3. 在 Outposts access points (Outposts 存取點) 區段中，選擇您要刪除的 Outposts 存取點。
4. 選擇 Delete (刪除)。
5. 確認刪除。

使用 AWS CLI

下列 AWS CLI 範例會刪除 Outposts 存取點。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control delete-access-point --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

新增或編輯存取點政策

存取點有 Amazon S3 on Outposts 對於透過該存取點進行的任何請求所套用的不同許可和網路控制。每個存取點都會強制執行自訂的存取點政策，該政策可結合附加至基礎儲存貯體的儲存貯體政策運作。如需詳細資訊，請參閱存取點。

下列主題說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 新增或編輯 S3 on Outposts 存取點的存取點政策 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要編輯存取點政策的 Outposts 儲存貯體。
4. 選擇 Outposts access points (Outposts 存取點) 索引標籤。
5. 在 Outposts access points (Outposts 存取點) 區段中，選取您要編輯其政策的存取點，然後選擇 Edit policy (編輯政策)。
6. 在 Outposts access point policy (Outposts 存取點政策) 區段中新增或編輯政策。如需詳細資訊，請參閱使用 S3 on Outposts 設定 IAM。

使用 AWS CLI

下列 AWS CLI 範例會在 Outposts 存取點上放置政策。

1. 將以下存取點政策儲存至 JSON 檔案。在此範例中，檔案命名為 `appolicy1.json`。以您自己的資訊取代 *user input placeholders*。

```
{
   "Version":"2012-10-17",
```

```
   "Id":"exampleAccessPointPolicy",
   "Statement":[
      {
         "Sid":"st1",
         "Effect":"Allow",
         "Principal":{
            "AWS":"123456789012"
         },
         "Action":"s3-outposts:*",
         "Resource":"arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point
      }
   ]
}
```

2. 提交 JSON 檔案以做為 `put-access-point-policy` CLI 命令的一部分。以您的資訊取代 *user input placeholders*。

```
aws s3control put-access-point-policy --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --policy file://appolicy1.json
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例在 Outposts 存取點上放置政策。

```
import com.amazonaws.services.s3control.model.*;

public void putAccessPointPolicy(String accessPointArn) {

    String policy = "{\"Version\":\"2012-10-17\",\"Id\":\"testAccessPointPolicy\",
\"Statement\":[{\"Sid\":\"st1\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"" +
 AccountId + "\"},\"Action\":\"s3-outposts:*\",\"Resource\":\"" + accessPointArn +
 "\"}]}";

    PutAccessPointPolicyRequest reqPutAccessPointPolicy = new
 PutAccessPointPolicyRequest()
            .withAccountId(AccountId)
            .withName(accessPointArn)
            .withPolicy(policy);
```

```
    PutAccessPointPolicyResult respPutAccessPointPolicy =
 s3ControlClient.putAccessPointPolicy(reqPutAccessPointPolicy);
    System.out.printf("PutAccessPointPolicy Response: %s%n",
 respPutAccessPointPolicy.toString());
    printWriter.printf("PutAccessPointPolicy Response: %s%n",
 respPutAccessPointPolicy.toString());

}
```

查看 S3 on Outposts 存取點政策

存取點有 Amazon S3 on Outposts 對於透過該存取點進行的任何請求所套用的不同許可和網路控制。每個存取點都會強制執行自訂的存取點政策，該政策可結合附加至基礎儲存貯體的儲存貯體政策運作。如需詳細資訊，請參閱存取點。

如需使用 S3 on Outposts 存取點的詳細資訊，請參閱 使用 S3 on Outposts 儲存貯體。

下列主題說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 來檢視 S3 on Outposts 存取點政策 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts access points (Outposts 存取點)。
3. 選擇您要編輯存取點政策的 Outposts 存取點。
4. 在Permissions (許可)標籤上，檢閱 S3 on Outposts 存取點政策。
5. 若要編輯存取點政策，請參閱 新增或編輯存取點政策。

使用 AWS CLI

下列 AWS CLI 範例取得 Outposts 存取點的政策。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control get-access-point-policy --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例取得 Outposts 存取點的政策。

```
import com.amazonaws.services.s3control.model.*;

public void getAccessPointPolicy(String accessPointArn) {

    GetAccessPointPolicyRequest reqGetAccessPointPolicy = new
 GetAccessPointPolicyRequest()
            .withAccountId(AccountId)
            .withName(accessPointArn);

    GetAccessPointPolicyResult respGetAccessPointPolicy =
 s3ControlClient.getAccessPointPolicy(reqGetAccessPointPolicy);
    System.out.printf("GetAccessPointPolicy Response: %s%n",
 respGetAccessPointPolicy.toString());
    printWriter.printf("GetAccessPointPolicy Response: %s%n",
 respGetAccessPointPolicy.toString());

}
```

使用 Amazon S3 on Outposts 端點

若要將請求路由至 Amazon S3 on Outpost 存取點，您必須建立和設定 S3 on Outposts 端點。若要建立端點，您需要與 Outposts 主要區域的服務連結有效連接。Outpost 上的每個 Virtual Private Cloud (VPC) 可以有一個相關聯的端點。如需端點配額的詳細資訊，請參閱 S3 on Outposts 網路需求。您必須建立端點，才能存取您的 Outposts 儲存貯體並執行物件操作。如需詳細資訊，請參閱端點。

建立端點後，您可以使用「狀態」欄位來了解端點狀態。若您的 Outposts 處於離線狀態，系統將回傳 CREATE_FAILED。您可以檢查服務連結的連線、刪除端點，然後在連線恢復後重試建立操作。如需其他錯誤代碼列表，請參閱下方內容。如需詳細資訊，請參閱端點。

API	Status	失敗原因錯誤代碼	訊息 - 失敗原因
`CreateEndpoint`	Create_Failed	OutpostNotReachable	因為與 Outposts 主要區域的服務連結連線失敗，所以無法建立端點。請檢查您的連線狀態、刪除端點，然後再試一次。
`CreateEndpoint`	Create_Failed	InternalError	由於內部錯誤，無法建立端點。請刪除端點然後重新建立一次。

API	Status	失敗原因錯誤代碼	訊息 - 失敗原因
DeleteEndpoint	Delete_Failed	OutpostNotReachable	因為與 Outposts 主要區域的服務連結連線失敗，所以無法刪除端點。請檢查您的連線狀態，然後再試一次。
DeleteEndpoint	Delete_Failed	InternalError	由於內部錯誤，無法刪除端點。請再試一次。

如需在 S3 on Outposts 上使用儲存貯體的詳細資訊，請參閱 使用 S3 on Outposts 儲存貯體。

以下幾節描述如何建立和管理 S3 on Outposts 的端點。

主題

- 在 Outpost 上建立端點
- 檢視 Amazon S3 on Outposts 端點上的清單
- 刪除 Amazon S3 on Outposts 端點

在 Outpost 上建立端點

若要將請求路由至 Amazon S3 on Outpost 存取點，您必須建立和設定 S3 on Outposts 端點。若要建立端點，您需要與 Outposts 主要區域的服務連結有效連接。Outpost 上的每個 Virtual Private Cloud (VPC) 可以有一個相關聯的端點。如需端點配額的詳細資訊，請參閱 S3 on Outposts 網路需求。您必須建立端點，才能存取您的 Outposts 儲存貯體並執行物件操作。如需詳細資訊，請參閱端點。

許可

如需建立端點所需許可的詳細資訊，請參閱 適用於 S3 on Outposts 端點的許可。

當您建立端點時，S3 on Outposts 也會在您的 AWS 帳戶中建立服務連結角色。如需詳細資訊，請參閱針對 Amazon S3 on Outposts 使用服務連結角色。

下列範例示範如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 建立 S3 on Outposts 端點 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台：//https：/https://console.aws.amazon.com/s3/.microsoft.com。

2. 在左側導覽窗格中，選擇 Outposts access points (Outposts 存取點)。
3. 選擇 Outposts endpoints (Outposts 端點) 索引標籤。
4. 選擇 Create Outposts endpoint (建立 Outposts 端點)。
5. 在 Outpost 下，選擇 Outpost 以建立此端點。
6. 在 VPC 下，選擇還沒有端點且符合 Ourposts 端點規則的 VPC。

   虛擬私有雲端 (VPC) 可讓您在定義的虛擬網路中啟動 AWS 資源。這個虛擬網路與您在資料中心中操作的傳統網路非常相似，且具備使用 可擴展基礎設施的優勢 AWS

   如果您沒有 VPC，請選擇 Create VPC (建立 VPC)。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的建立僅限虛擬私有雲端 (VPC) 使用的存取點。
7. 選擇 Create Outposts endpoint (建立 Outposts 端點)。

使用 AWS CLI

Example

下列 AWS CLI 範例會使用 VPC 資源存取類型，為 Outpost 建立端點。VPC 衍生自子網路。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

下列 AWS CLI 範例使用客戶擁有的 IP 地址集區 (CoIP 集區） 存取類型，為 Outpost 建立端點。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --
customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

使用適用於 Java 的 AWS 開發套件

Example

下列適用於 Java 的開發套件範例建立 Outposts 的端點。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
```

```
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.CreateEndpointRequest;
import com.amazonaws.services.s3outposts.model.CreateEndpointResult;

public void createEndpoint() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
                .standard().build();

    CreateEndpointRequest createEndpointRequest = new CreateEndpointRequest()
                .withOutpostId("op-0d79779cef3c30a40")
                .withSubnetId("subnet-8c7a57c5")
                .withSecurityGroupId("sg-ab19e0d1")
                .withAccessType("CustomerOwnedIp")
                .withCustomerOwnedIpv4Pool("ipv4pool-coip-12345678901234567");
    // Use .withAccessType and .withCustomerOwnedIpv4Pool only when the access type is
    // customer-owned IP address pool (CoIP pool)
    CreateEndpointResult createEndpointResult =
 s3OutpostsClient.createEndpoint(createEndpointRequest);
    System.out.println("Endpoint is created and its ARN is " +
 createEndpointResult.getEndpointArn());
}
```

檢視 Amazon S3 on Outposts 端點上的清單

若要將請求路由至 Amazon S3 on Outpost 存取點，您必須建立和設定 S3 on Outposts 端點。若要建立端點，您需要與 Outposts 主要區域的服務連結有效連接。Outpost 上的每個 Virtual Private Cloud (VPC) 可以有一個相關聯的端點。如需端點配額的詳細資訊，請參閱 S3 on Outposts 網路需求。您必須建立端點，才能存取您的 Outposts 儲存貯體並執行物件操作。如需詳細資訊，請參閱端點。

下列範例示範如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 傳回 S3 on Outposts 端點的清單 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts access points (Outposts 存取點)。
3. 在 Outposts access points (Outposts 存取點)頁面上，選擇 Outposts endpoints (Outposts 端點)標籤。
4. 在 Outposts endpoints (Outposts 端點)中，您可以查看 S3 on Outposts 端點清單。

使用 AWS CLI

下列 AWS CLI 範例會列出與您帳戶相關聯之 AWS Outposts 資源的端點。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 list-endpoints。

```
aws s3outposts list-endpoints
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例列出了 Outpost 的端點。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 ListEndpoints。

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.ListEndpointsRequest;
import com.amazonaws.services.s3outposts.model.ListEndpointsResult;

public void listEndpoints() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
                .standard().build();

    ListEndpointsRequest listEndpointsRequest = new ListEndpointsRequest();
    ListEndpointsResult listEndpointsResult =
 s3OutpostsClient.listEndpoints(listEndpointsRequest);
    System.out.println("List endpoints result is " + listEndpointsResult);
}
```

刪除 Amazon S3 on Outposts 端點

若要將請求路由至 Amazon S3 on Outpost 存取點，您必須建立和設定 S3 on Outposts 端點。若要建立端點，您需要與 Outposts 主要區域的服務連結有效連接。Outpost 上的每個 Virtual Private Cloud (VPC) 可以有一個相關聯的端點。如需端點配額的詳細資訊，請參閱 S3 on Outposts 網路需求。您必須建立端點，才能存取您的 Outposts 儲存貯體並執行物件操作。如需詳細資訊，請參閱端點。

下列範例說明如何使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 刪除 S3 on Outposts 端點 適用於 Java 的 AWS SDK。

使用 S3 主控台

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts access points (Outposts 存取點)。

3. 在 Outposts access points (Outposts 存取點)頁面上，選擇 Outposts endpoints (Outposts 端點)標籤。
4. 在 Outposts 端點下，選擇您要刪除的端點，然後選擇Delete (刪除)。

使用 AWS CLI

下列 AWS CLI 範例會刪除 Outpost 的端點。若要執行此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3outposts delete-endpoint --endpoint-id example-endpoint-id --outpost-
id op-01ac5d28a6a232904
```

使用適用於 Java 的 AWS 開發套件

下列適用於 Java 的開發套件範例刪除 Outpost 的端點。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
import com.amazonaws.arn.Arn;
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.DeleteEndpointRequest;

public void deleteEndpoint(String endpointArnInput) {
    String outpostId = "op-01ac5d28a6a232904";
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
                .standard().build();

    Arn endpointArn = Arn.fromString(endpointArnInput);
    String[] resourceParts = endpointArn.getResource().getResource().split("/");
    String endpointId = resourceParts[resourceParts.length - 1];
    DeleteEndpointRequest deleteEndpointRequest = new DeleteEndpointRequest()
                .withEndpointId(endpointId)
                .withOutpostId(outpostId);
    s3OutpostsClient.deleteEndpoint(deleteEndpointRequest);
    System.out.println("Endpoint with id " + endpointId + " is deleted.");
}
```

使用 S3 on Outposts 物件

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (0UTP0STS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。

物件是存放在 Amazon S3 on Outposts 中的基本實體。每個物件都包含在儲存貯體中。您必須使用存取點來存取 Outpost 儲存貯體中的任何物件。針對物件操作指定儲存貯體時，您可以使用存取點 Amazon Resource Name (ARN) 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

下列範例顯示 S3 on Outposts 存取點的 ARN 格式，其中包含 Outpost 所在區域的 AWS 區域 程式碼、 AWS 帳戶 ID、Outpost ID 和存取點名稱：

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

物件 ARNs 使用下列格式，其中包括 AWS 區域 Outpost 所在的 、 AWS 帳戶 ID、Outpost ID、儲存貯體名稱和物件金鑰：

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket1/object/myobject
```

對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

主題

- 將物件上傳至 S3 on Outposts 儲存貯體

- 使用 在 Amazon S3 on Outposts 儲存貯體中複製物件 適用於 Java 的 AWS SDK
- 從 Amason S3 on Outposts 儲存貯體取得物件
- 列出 Amazon S3 on Outposts 儲存貯體中的物件
- 刪除 Amazon S3 on Outposts 儲存貯體中的物件
- 使用 HeadBucket 判斷 S3 on Outposts 儲存貯體是否存在，並且您是否擁有存取許可
- 使用適用於 Java 的開發套件執行和管理分段上傳
- 使用適用於 S3 on OutOutposts 的預先簽章 URL
- 搭配使用 Amazon S3 on Outposts 和本機 Amazon EMR on Outposts
- 授權與身分驗證快取

將物件上傳至 S3 on Outposts 儲存貯體

物件是存放在 Amazon S3 on Outposts 中的基本實體。每個物件都包含在儲存貯體中。您必須使用存取點來存取 Outpost 儲存貯體中的任何物件。針對物件操作指定儲存貯體時，您可以使用存取點 Amazon Resource Name (ARN) 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

下列範例顯示 S3 on Outposts 存取點的 ARN 格式，其中包含 Outpost 所在區域的 AWS 區域 程式碼、 AWS 帳戶 ID、Outpost ID 和存取點名稱：

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

下列 AWS CLI 和 適用於 Java 的 AWS SDK 範例示範如何使用存取點將物件上傳至 S3 on Outposts 儲存貯體。

AWS CLI

Example

下列範例使用 AWS CLI將名稱為 `sample-object.xml` 的物件放置在 S3 on Outposts 儲存貯體 (`s3-outposts:PutObject`) 上。若要執行此命令，請以您自己的資訊取代每個 *`user input placeholder`*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 put-object。

```
aws s3api put-object --bucket arn:aws:s3-
outposts:Region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --key sample-object.xml --body sample-object.xml
```

SDK for Java

Example

下列範例使用適用於 Java 的開發套件，將物件放置在 S3 on Outposts 儲存貯體。若要使用此範例，請以您自己的資訊取代每個 *`user input placeholder`*。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

import java.io.File;

public class PutObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
        String fileName = "*** Path to file to upload ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
```

```
                    .build();

            // Upload a text string as a new object.
            s3Client.putObject(accessPointArn, stringObjKeyName, "Uploaded String
 Object");

            // Upload a file as a new object with ContentType and title specified.
            PutObjectRequest request = new PutObjectRequest(accessPointArn,
 fileObjKeyName, new File(fileName));
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentType("plain/text");
            metadata.addUserMetadata("title", "someTitle");
            request.setMetadata(metadata);
            s3Client.putObject(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

使用 在 Amazon S3 on Outposts 儲存貯體中複製物件 適用於 Java 的 AWS SDK

物件是存放在 Amazon S3 on Outposts 中的基本實體。每個物件都包含在儲存貯體中。您必須使用存取點來存取 Outpost 儲存貯體中的任何物件。針對物件操作指定儲存貯體時，您可以使用存取點 Amazon Resource Name (ARN) 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

下列範例顯示 S3 on Outposts 存取點的 ARN 格式，其中包含 Outpost 所在區域的 AWS 區域 程式碼、 AWS 帳戶 ID、Outpost ID 和存取點名稱：

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

下列範例示範如何使用 適用於 Java 的 AWS SDK複製 S3 on Outposts 儲存貯體中的物件。

使用適用於 Java 的 AWS 開發套件

下列 S3 on Outposts 範例使用適用於 Java 的開發套件，將物件複製到同一儲存貯體中的新物件。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

public class CopyObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String sourceKey = "*** Source object key ***";
        String destinationKey = "*** Destination object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Copy the object into a new object in the same bucket.
            CopyObjectRequest copyObjectRequest = new CopyObjectRequest(accessPointArn,
 sourceKey, accessPointArn, destinationKey);
            s3Client.copyObject(copyObjectRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
```

```
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

從 Amason S3 on Outposts 儲存貯體取得物件

物件是存放在 Amazon S3 on Outposts 中的基本實體。每個物件都包含在儲存貯體中。您必須使用存取點來存取 Outpost 儲存貯體中的任何物件。針對物件操作指定儲存貯體時，您可以使用存取點 Amazon Resource Name (ARN) 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

下列範例顯示 S3 on Outposts 存取點的 ARN 格式，其中包含 Outpost 所在區域的 AWS 區域 程式碼、 AWS 帳戶 ID、Outpost ID 和存取點名稱：

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

下列範例示範如何使用 AWS Command Line Interface (AWS CLI) 和 適用於 Java 的 AWS SDK下載 (取得) 物件。

使用 AWS CLI

下列範例使用 AWS CLI從 S3 on Outposts 儲存貯體 (`s3-outposts:GetObject`) 取得名稱為 `sample-object.xml` 的物件。若要執行此命令，請以您自己的資訊取代每個 *`user input placeholder`*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 get-object。

```
aws s3api get-object --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point --key testkey sample-object.xml
```

使用適用於 Java 的 AWS 開發套件

下列 S3 on Outposts 範例使用適用於 Java 的開發套件取得物件。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。如需詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的 GetObject。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.ResponseHeaderOverrides;
import com.amazonaws.services.s3.model.S3Object;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class GetObject {
    public static void main(String[] args) throws IOException {
        String accessPointArn = "*** access point ARN ***";
        String key = "*** Object key ***";

        S3Object fullObject = null, objectPortion = null, headerOverrideObject = null;
        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Get an object and print its contents.
            System.out.println("Downloading an object");
            fullObject = s3Client.getObject(new GetObjectRequest(accessPointArn, key));
            System.out.println("Content-Type: " +
 fullObject.getObjectMetadata().getContentType());
            System.out.println("Content: ");
            displayTextInputStream(fullObject.getObjectContent());

            // Get a range of bytes from an object and print the bytes.
```

```
            GetObjectRequest rangeObjectRequest = new GetObjectRequest(accessPointArn,
key)
                    .withRange(0, 9);
            objectPortion = s3Client.getObject(rangeObjectRequest);
            System.out.println("Printing bytes retrieved.");
            displayTextInputStream(objectPortion.getObjectContent());

            // Get an entire object, overriding the specified response headers, and
print the object's content.
            ResponseHeaderOverrides headerOverrides = new ResponseHeaderOverrides()
                    .withCacheControl("No-cache")
                    .withContentDisposition("attachment; filename=example.txt");
            GetObjectRequest getObjectRequestHeaderOverride = new
GetObjectRequest(accessPointArn, key)
                    .withResponseHeaders(headerOverrides);
            headerOverrideObject = s3Client.getObject(getObjectRequestHeaderOverride);
            displayTextInputStream(headerOverrideObject.getObjectContent());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        } finally {
            // To ensure that the network connection doesn't remain open, close any
open input streams.
            if (fullObject != null) {
                fullObject.close();
            }
            if (objectPortion != null) {
                objectPortion.close();
            }
            if (headerOverrideObject != null) {
                headerOverrideObject.close();
            }
        }
    }

    private static void displayTextInputStream(InputStream input) throws IOException {
        // Read the text input stream one line at a time and display each line.
        BufferedReader reader = new BufferedReader(new InputStreamReader(input));
        String line = null;
```

```
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }
        System.out.println();
    }
}
```

列出 Amazon S3 on Outposts 儲存貯體中的物件

物件是存放在 Amazon S3 on Outposts 中的基本實體。每個物件都包含在儲存貯體中。您必須使用存取點來存取 Outpost 儲存貯體中的任何物件。針對物件操作指定儲存貯體時，您可以使用存取點 Amazon Resource Name (ARN) 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

下列範例顯示 S3 on Outposts 存取點的 ARN 格式，其中包含 Outpost 所在區域的 AWS 區域 程式碼、 AWS 帳戶 ID、Outpost ID 和存取點名稱：

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

> **Note**
>
> 對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

下列範例說明如何使用 AWS CLI 和 列出 S3 on Outposts 儲存貯體中的物件 適用於 Java 的 AWS SDK。

使用 AWS CLI

下列範例使用 AWS CLI列出 S3 on Outposts 儲存貯體 (`s3-outposts:ListObjectsV2`) 中的物件。若要執行此命令，請以您自己的資訊取代每個 *`user input placeholder`*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 list-objects-v2。

```
aws s3api list-objects-v2 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

> **Note**
>
> 透過 AWS SDK 將此動作與 Amazon S3 on Outposts 搭配使用時，您可以提供 Outposts 存取點 ARN 取代儲存貯體名稱，格式如下：`arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-Outposts-Access-Point`。如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

使用適用於 Java 的 AWS 開發套件

下列 S3 on Outposts 範例使用適用於 Java 的開發套件，在儲存貯體中列出物件。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。

> **Important**
>
> 此範例使用 ListObjectsV2，這是 `ListObjects` API 操作的最新修訂版。建議您使用此修訂版本後的 API 操作進行應用程式進行開發。為了回溯相容性，Amazon S3 會繼續支援此 API 操作的舊版本。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsV2Request;
import com.amazonaws.services.s3.model.ListObjectsV2Result;
import com.amazonaws.services.s3.model.S3ObjectSummary;

public class ListObjectsV2 {

    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
```

```
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            System.out.println("Listing objects");

            // maxKeys is set to 2 to demonstrate the use of
            // ListObjectsV2Result.getNextContinuationToken()
            ListObjectsV2Request req = new
 ListObjectsV2Request().withBucketName(accessPointArn).withMaxKeys(2);
            ListObjectsV2Result result;

            do {
                result = s3Client.listObjectsV2(req);

                for (S3ObjectSummary objectSummary : result.getObjectSummaries()) {
                    System.out.printf(" - %s (size: %d)\n", objectSummary.getKey(),
 objectSummary.getSize());
                }
                // If there are more than maxKeys keys in the bucket, get a
 continuation token
                // and list the next objects.
                String token = result.getNextContinuationToken();
                System.out.println("Next Continuation Token: " + token);
                req.setContinuationToken(token);
            } while (result.isTruncated());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

刪除 Amazon S3 on Outposts 儲存貯體中的物件

物件是存放在 Amazon S3 on Outposts 中的基本實體。每個物件都包含在儲存貯體中。您必須使用存取點來存取 Outpost 儲存貯體中的任何物件。針對物件操作指定儲存貯體時，您可以使用存取點 Amazon Resource Name (ARN) 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

下列範例顯示 S3 on Outposts 存取點的 ARN 格式，其中包含 Outpost 所在區域的 AWS 區域 程式碼、 AWS 帳戶 ID、Outpost ID 和存取點名稱：

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

下列範例說明如何使用 AWS Command Line Interface (AWS CLI) 和 刪除 S3 on Outposts 儲存貯體中的單一物件或多個物件 適用於 Java 的 AWS SDK。

使用 AWS CLI

下列範例示範如何從 S3 on Outposts 儲存貯體中刪除單個物件或多個物件。

delete-object

下列範例使用 AWS CLI從 S3 on Outposts 儲存貯體 (`s3-outposts:DeleteObject`) 刪除名稱為 `sample-object.xml` 的物件。若要執行此命令，請以您自己的資訊取代每個 *`user input placeholder`*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 delete-object。

```
aws s3api delete-object --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --key sample-object.xml
```

delete-objects

下列範例使用 AWS CLI從 S3 on Outposts 儲存貯體 (`s3-outposts:DeleteObject`) 刪除名稱為 `sample-object.xml` 和 `test1.text` 的兩個物件。若要執行此命令，請以您自己的資訊取代每個 *`user input placeholder`*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 delete-objects。

```
aws s3api delete-objects --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --delete file://delete.json

delete.json
{
  "Objects": [
    {
      "Key": "test1.txt"
    },
    {
      "Key": "sample-object.xml"
    }
  ],
  "Quiet": false
}
```

使用適用於 Java 的 AWS 開發套件

下列範例示範如何從 S3 on Outposts 儲存貯體中刪除單個物件或多個物件。

DeleteObject

下列 S3 on Outposts 範例使用適用於 Java 的開發套件，在儲存貯體中刪除物件。若要使用此範例，請為 Outpost 指定存取點 ARN，並為您要刪除的物件指定金鑰名稱。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的DeleteObject。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;

public class DeleteObject {
```

```
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** key name ****";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            s3Client.deleteObject(new DeleteObjectRequest(accessPointArn, keyName));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

DeleteObjects

下列 S3 on Outposts 範例使用適用於 Java 的開發套件，在儲存貯體中上傳，然後刪除物件。若要使用此範例，請為 Outpost 指定存取點 ARN。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 DeleteObjects。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectsRequest;
import com.amazonaws.services.s3.model.DeleteObjectsRequest.KeyVersion;
import com.amazonaws.services.s3.model.DeleteObjectsResult;

import java.util.ArrayList;

public class DeleteObjects {
```

```
    public static void main(String[] args) {
       String accessPointArn = "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point";


        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Upload three sample objects.
            ArrayList<KeyVersion> keys = new ArrayList<KeyVersion>();
            for (int i = 0; i < 3; i++) {
                String keyName = "delete object example " + i;
                s3Client.putObject(accessPointArn, keyName, "Object number " + i + "
 to be deleted.");
                keys.add(new KeyVersion(keyName));
            }
            System.out.println(keys.size() + " objects successfully created.");

            // Delete the sample objects.
            DeleteObjectsRequest multiObjectDeleteRequest = new
 DeleteObjectsRequest(accessPointArn)
                    .withKeys(keys)
                    .withQuiet(false);

            // Verify that the objects were deleted successfully.
            DeleteObjectsResult delObjRes =
 s3Client.deleteObjects(multiObjectDeleteRequest);
            int successfulDeletes = delObjRes.getDeletedObjects().size();
            System.out.println(successfulDeletes + " objects successfully
 deleted.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
```

```
        }
    }
}
```

使用 HeadBucket 判斷 S3 on Outposts 儲存貯體是否存在，並且您是否擁有存取許可

物件是存放在 Amazon S3 on Outposts 中的基本實體。每個物件都包含在儲存貯體中。您必須使用存取點來存取 Outpost 儲存貯體中的任何物件。針對物件操作指定儲存貯體時，您可以使用存取點 Amazon Resource Name (ARN) 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

下列範例顯示 S3 on Outposts 存取點的 ARN 格式，其中包含 Outpost 所在區域的 AWS 區域 程式碼、 AWS 帳戶 ID、Outpost ID 和存取點名稱：

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

如需 S3 on Outposts ARN 的詳細資訊，請參閱 適用於 S3 on Outposts 的資源 ARN。

> **Note**
>
> 對於 Amazon S3 on Outposts，物件資料始終存放在 Outpost 上。當 AWS 安裝 Outpost 機架時，您的資料會保留在 Outpost 的本機，以符合資料備援需求。您的物件永遠不會離開您的 Outpost，也不會在 AWS 區域中。由於 AWS Management Console 是在 區域中託管，因此您無法使用 主控台來上傳或管理 Outpost 中的物件。不過，您可以使用 REST API、 AWS Command Line Interface (AWS CLI) 和 AWS SDKs 透過存取點上傳和管理物件。

以下 AWS Command Line Interface (AWS CLI) 和 適用於 Java 的 AWS SDK 範例說明如何使用 HeadBucket API 操作來判斷 Amazon S3 on Outposts 儲存貯體是否存在，以及您是否具有存取它的許可。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 HeadBucket。

使用 AWS CLI

下列 S3 on Outposts AWS CLI 範例使用 `head-bucket`命令來判斷儲存貯體是否存在，而且您具有存取儲存貯體的許可。若要執行此命令，請以您自己的資訊取代每個 *`user input placeholder`*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 head-bucket。

```
aws s3api head-bucket --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

使用適用於 Java 的 AWS 開發套件

下列 S3 on Outposts 範例顯示如何判斷儲存貯體是否存在，以及您是否有存取許可。若要使用此範例，請為 Outpost 指定存取點 ARN。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 HeadBucket。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.HeadBucketRequest;

public class HeadBucket {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            s3Client.headBucket(new HeadBucketRequest(accessPointArn));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

使用適用於 Java 的開發套件執行和管理分段上傳

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 資源上建立 S3 儲存貯體，並為需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署存放和擷取物件。您可以透過 AWS Management Console、 AWS Command Line Interface (AWS CLI)、 AWS SDKs 或 REST API 使用 S3 on Outposts。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

下列範例示範如何搭配 使用 S3 on Outposts 適用於 Java 的 AWS SDK ，以執行和管理分段上傳。

主題

- 在 S3 on Outposts 儲存貯體中執行物件的分段上傳
- 使用分段上傳在 S3 on Outposts 儲存貯體中複製大型物件
- 列出 S3 on Outposts 儲存貯體中物件的片段
- 擷取 S3 on Outposts 儲存貯體中進行中的分段上傳清單

在 S3 on Outposts 儲存貯體中執行物件的分段上傳

下列 S3 on Outposts 範例使用適用於 Java 的開發套件，在 Outposts 儲存貯體啟動、上傳和完成物件的分段上傳。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的使用分段上傳來上傳物件。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
```

```
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
 InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
 s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
 GetObjectMetadataRequest(accessPointArn, sourceObjectKey);
            ObjectMetadata metadataResult =
 s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

                // Copy this part.
                CopyPartRequest copyRequest = new CopyPartRequest()
                        .withSourceBucketName(accessPointArn)
                        .withSourceKey(sourceObjectKey)
                        .withDestinationBucketName(accessPointArn)
                        .withDestinationKey(destObjectKey)
                        .withUploadId(initResult.getUploadId())
                        .withFirstByte(bytePosition)
                        .withLastByte(lastByte)
                        .withPartNumber(partNum++);
                copyResponses.add(s3Client.copyPart(copyRequest));
                bytePosition += partSize;
            }
```

```
            // Complete the upload request to concatenate all uploaded parts and make
the copied object available.
            CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
                    accessPointArn,
                    destObjectKey,
                    initResult.getUploadId(),
                    getETags(copyResponses));
            s3Client.completeMultipartUpload(completeRequest);
            System.out.println("Multipart copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    // This is a helper function to construct a list of ETags.
    private static List<PartETag> getETags(List<CopyPartResult> responses) {
        List<PartETag> etags = new ArrayList<PartETag>();
        for (CopyPartResult response : responses) {
            etags.add(new PartETag(response.getPartNumber(), response.getETag()));
        }
        return etags;
    }
```

使用分段上傳在 S3 on Outposts 儲存貯體中複製大型物件

下列 S3 on Outposts 範例使用適用於 Java 的開發套件在儲存貯體中複製物件。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;
```

```
public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
 InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
 s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
 GetObjectMetadataRequest(accessPointArn, sourceObjectKey);
            ObjectMetadata metadataResult =
 s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

                // Copy this part.
                CopyPartRequest copyRequest = new CopyPartRequest()
                        .withSourceBucketName(accessPointArn)
                        .withSourceKey(sourceObjectKey)
                        .withDestinationBucketName(accessPointArn)
                        .withDestinationKey(destObjectKey)
```

```
                        .withUploadId(initResult.getUploadId())
                        .withFirstByte(bytePosition)
                        .withLastByte(lastByte)
                        .withPartNumber(partNum++);
                copyResponses.add(s3Client.copyPart(copyRequest));
                bytePosition += partSize;
            }

            // Complete the upload request to concatenate all uploaded parts and make
 the copied object available.
            CompleteMultipartUploadRequest completeRequest = new
 CompleteMultipartUploadRequest(
                    accessPointArn,
                    destObjectKey,
                    initResult.getUploadId(),
                    getETags(copyResponses));
            s3Client.completeMultipartUpload(completeRequest);
            System.out.println("Multipart copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    // This is a helper function to construct a list of ETags.
    private static List<PartETag> getETags(List<CopyPartResult> responses) {
        List<PartETag> etags = new ArrayList<PartETag>();
        for (CopyPartResult response : responses) {
            etags.add(new PartETag(response.getPartNumber(), response.getETag()));
        }
        return etags;
    }
}
```

列出 S3 on Outposts 儲存貯體中物件的片段

下列 S3 on Outposts 範例使用適用於 Java 的開發套件，在儲存貯體中列出物件的片段。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.List;

public class ListParts {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** Key name ***";
        String uploadId = "*** Upload ID ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            ListPartsRequest listPartsRequest = new ListPartsRequest(accessPointArn,
 keyName, uploadId);
            PartListing partListing = s3Client.listParts(listPartsRequest);
            List<PartSummary> partSummaries = partListing.getParts();

            System.out.println(partSummaries.size() + " multipart upload parts");
            for (PartSummary p : partSummaries) {
                System.out.println("Upload part: Part number = \"" + p.getPartNumber()
 + "\", ETag = " + p.getETag());
            }

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
```

```
}
```

擷取 S3 on Outposts 儲存貯體中進行中的分段上傳清單

下列 S3 on Outposts 範例顯示如何使用適用於 Java 的開發套件，從 Outposts 儲存貯體擷取進行中的分段上傳清單。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListMultipartUploadsRequest;
import com.amazonaws.services.s3.model.MultipartUpload;
import com.amazonaws.services.s3.model.MultipartUploadListing;

import java.util.List;

public class ListMultipartUploads {
    public static void main(String[] args) {
                String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Retrieve a list of all in-progress multipart uploads.
            ListMultipartUploadsRequest allMultipartUploadsRequest = new
 ListMultipartUploadsRequest(accessPointArn);
            MultipartUploadListing multipartUploadListing =
 s3Client.listMultipartUploads(allMultipartUploadsRequest);
            List<MultipartUpload> uploads =
 multipartUploadListing.getMultipartUploads();

            // Display information about all in-progress multipart uploads.
            System.out.println(uploads.size() + " multipart upload(s) in progress.");
            for (MultipartUpload u : uploads) {
                System.out.println("Upload in progress: Key = \"" + u.getKey() + "\",
 id = " + u.getUploadId());
            }
```

```
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

使用適用於 S3 on OutOutposts 的預先簽章 URL

若要授予對存放在本機 Outpost 上物件的有限時間存取權限，而不會更新儲存貯體政策，您可以使用預先簽章 URL。使用預先簽章 URL，身為儲存貯體擁有者的您可以與虛擬私有雲端 (VPC) 中的個人共享物件，或授予他們上傳或刪除物件的能力。

當您使用 AWS SDKs或 AWS Command Line Interface (AWS CLI) 建立預先簽章的 URL 時，您可以將 URL 與特定動作建立關聯。您也可以選擇自訂到期時間 (最低 1 秒，最高 7 天) 來授予預先簽章 URL 有限時間的存取權。當您共用預先簽章 URL 時，VPC 中的個人可以執行內嵌在 URL 中的動作，如同原始簽章使用者一樣。當 URL 到達到期時間時，該 URL 就會過期且再也無法運作。

限制預先簽章的 URL 功能

預先簽章的 URL 的功能，受到建立它的使用者許可所限制。實質上，預先簽章的 URL 是一種承載符記，可為擁有這些網址的客戶授與存取權。因此，我們建議您妥善保護它們。

AWS Signature 第 4 版 (SigV4)

若要在使用 AWS Signature 第 4 版 (SigV4) 驗證預先簽章的 URL 請求時強制執行特定行為，您可以在儲存貯體政策和存取點政策中使用條件金鑰。例如，您可以建立儲存貯體政策，使用 `s3-outposts:signatureAge` 條件來拒絕任何 `example-outpost-bucket` 儲存貯體中物件上的 Amazon S3 on Outposts 預先簽章 URL 請求 (如果簽章超過 10 分鐘)。若要使用此範例，請以您自己的資訊取代 *`user input placeholders`*。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "Deny a presigned URL request if the signature is more than 10
 minutes old",
            "Effect": "Deny",
            "Principal": {"AWS":"444455556666"},
            "Action": "s3-outposts:*",
            "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
            "Condition": {
                "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
                "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
            }
        }
    ]
}
```

如需取得可用來強制執行特定行為 (在使用第 4 版簽署程序驗證預先簽章的 URL 請求時) 的條件金鑰和其他政策範例清單，請參閱 AWS Signature 第 4 版 (SigV4) 身分驗證特定政策金鑰。

網路路徑限制

如果您想要限制使用預先簽章 URL 和對特定網路路徑的所有 S3 on Outposts 存取權，您可以撰寫需要特定網路路徑的政策。若要對進行呼叫的 IAM 主體設定限制，您可以使用身分型 AWS Identity and Access Management (IAM) 政策 （ 例如使用者、群組或角色政策 ）。若要在 S3 on Outposts 資源上設定的限制，您可以使以資源型政策 (例如儲存貯體和存取點政策)。

IAM 主體的網路路徑限制需要這些憑證的使用者從指定的網路發出請求。儲存貯體或存取點上的限制要求所有對該資源的請求都來自指定網路。這些限制也適用於預先簽章的 URL 案例之外。

您使用的 IAM 全域條件取決於端點類型。如果您正在使用 S3 on Outposts 的公有端點，請使用 `aws:SourceIp`。如果您正在使用 S3 on Outposts 的 VPC 端點，請使用 `aws:SourceVpc` 或 `aws:SourceVpce`。

下列 IAM 政策陳述式要求委託人 AWS 只能從指定的網路範圍存取 。由於此政策聲明，所有存取均必須源自該範圍。這包含某人使用 S3 on Outposts 預先簽章 URL 的情況。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
{
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
```

```
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

如需使用 aws:SourceIP AWS 全域條件金鑰將 S3 on Outposts 儲存貯體的存取限制在特定網路範圍的範例儲存貯體政策，請參閱 使用 S3 on Outposts 設定 IAM。

誰可以建立預先簽章的 URL

任何具備有效安全憑證的使用者，均可建立預先簽章的 URL。但為了讓 VPC 中的使用者能順利存取物件，預先簽章的 URL 必須由有權執行預先簽章的 URL 做為基礎之操作的人員來建立。

您可以使用下列憑證來建立預先簽章 URL：

- IAM 執行個體設定檔 - 有效期限最長 6 小時。
- AWS Security Token Service - 以 AWS 帳戶 根使用者或 IAM 使用者憑證等永久憑證簽章時，有效期限最長 36 小時。
- IAM 使用者 – 當您使用 AWS Signature 第 4 版時，有效期最長為 7 天。

  若要建立有效期限最長 7 天的預先簽章 URL，請先將 IAM 使用者憑證 (存取金鑰和私密金鑰) 委派給您在使用的 SDK。然後，使用 AWS Signature 第 4 版產生預先簽章的 URL。

> **Note**
>
> - 如果使用暫時字符建立了預先簽章的 URL，則 URL 會在字符過期時過期，即使您使用較晚的過期時間建立 URL 亦然。
> - 由於預先簽章 URL 會將 S3 on Outposts 儲存貯體的存取權授予擁有 URL 的任何人，因此我們建議您妥善保護這些 URL。如需保護預先簽章的 URL 的詳細資訊，請參閱限制預先簽章的 URL 功能。

S3 on Outposts 何時檢查預先簽章的 URL 中的到期日期和時間？

S3 on Outposts 會在發出 HTTP 請求時，檢查已簽署 URL 的到期日期和時間。例如，如果用戶端在到期前一刻才開始下載大型檔案，則即使在下載期間過期了，下載也會繼續。然而，如果連線中斷並且用戶端在到期時間過後嘗試重新啟動下載，則下載會失敗。

如需使用預先簽章 URL 來共享或上傳物件的詳細資訊，請參閱下列主題。

主題

使用預先簽章的 URL 來共用物件

若要授予對存放在本機 Outpost 上物件的有限時間存取權限，而不會更新儲存貯體政策，您可以使用預先簽章 URL。使用預先簽章 URL，身為儲存貯體擁有者的您可以與虛擬私有雲端 (VPC) 中的個人共享物件，或授予他們上傳或刪除物件的能力。

當您使用 AWS SDKs或 AWS Command Line Interface (AWS CLI) 建立預先簽章的 URL 時，您可以將 URL 與特定動作建立關聯。您也可以選擇自訂到期時間 (最低 1 秒，最高 7 天) 來授予預先簽章 URL 有限時間的存取權。當您共用預先簽章 URL 時，VPC 中的個人可以執行內嵌在 URL 中的動作，如同原始簽章使用者一樣。當 URL 到達到期時間時，該 URL 就會過期且再也無法運作。

當您建立預先簽章 URL 時，必須提供安全憑證，然後指定下列項目：

- 適用於 Amazon S3 on Outposts 儲存貯體的存取點 Amazon Resource Name (ARN)
- 物件索引鍵
- HTTP 方法 (GET 用於下載物件)
- 過期日期和時間

預先簽章 URL 僅在指定的期間內有效。也就是說，您必須在到期日期和時間之前開始 URL 所允許的操作。您可以多次使用預先簽章 URL，直到到期日期和時間為止。如果使用暫時字符建立了預先簽章的 URL，那麼 URL 會在字符過期時過期，即使您使用較晚的過期時間建立 URL 亦然。

虛擬私有雲端 (VPC) 中可存取預先簽章 URL 的使用者可以存取物件。例如，若儲存貯體中有一段影片且儲存貯體與物件皆為私有，即可透過產生預先簽章的 URL 來與其他人分享這段影片。由於預先簽章 URL 會將 S3 on Outposts 儲存貯體的存取權授予擁有 URL 的任何人，因此我們建議您妥善保護這些 URL。如需有關保護預先簽署 URL 的詳細資訊，請參閱限制預先簽章的 URL 功能。

任何具備有效安全憑證的使用者，均可建立預先簽章的 URL。然而，只有具備許可執行作為預先簽章 URL 基礎操作的人員，才能建立預先簽章 URL。如需詳細資訊，請參閱誰可以建立預先簽章的 URL。

您可以使用 AWS SDK 與 AWS CLI來產生預先簽章 URL，以在 S3 on Outposts 儲存貯體中共享物件。如需詳細資訊，請參閱下列範例。

使用 AWS SDKs

您可以使用 AWS SDKs來產生預先簽章的 URL，以便提供給其他人，讓他們可以擷取物件。

> **Note**
>
> 當您使用 AWS SDKs產生預先簽章的 URL 時，預先簽章的 URL 最長過期時間為建立時間起算 7 天。

Java

Example

以下範例會產生預先簽章的 URL，您可以將其提供給其他人，讓他們可以從 S3 on Outposts 儲存貯體擷取物件。如需詳細資訊，請參閱使用適用於 S3 on OutOutposts 的預先簽章 URL。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.HttpMethod;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GeneratePresignedUrlRequest;

import java.io.IOException;
import java.net.URL;
import java.time.Instant;

public class GeneratePresignedURL {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String accessPointArn = "*** access point ARN ***";
        String objectKey = "*** object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
```

```
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();

            // Set the presigned URL to expire after one hour.
            java.util.Date expiration = new java.util.Date();
            long expTimeMillis = Instant.now().toEpochMilli();
            expTimeMillis += 1000 * 60 * 60;
            expiration.setTime(expTimeMillis);

            // Generate the presigned URL.
            System.out.println("Generating pre-signed URL.");
            GeneratePresignedUrlRequest generatePresignedUrlRequest =
                    new GeneratePresignedUrlRequest(accessPointArn, objectKey)
                            .withMethod(HttpMethod.GET)
                            .withExpiration(expiration);
            URL url = s3Client.generatePresignedUrl(generatePresignedUrlRequest);

            System.out.println("Pre-Signed URL: " + url.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't
 process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Example

以下範例會產生預先簽章的 URL，您可以將其提供給其他人，讓他們可以從 S3 on Outposts 儲存貯體擷取物件。如需詳細資訊，請參閱使用適用於 S3 on OutOutposts 的預先簽章 URL。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
```

```
namespace Amazon.DocSamples.S3
{
    class GenPresignedURLTest
    {
        private const string accessPointArn = "*** access point ARN ***";
        private const string objectKey = "*** object key ***";
        // Specify how long the presigned URL lasts, in hours.
        private const double timeoutDuration = 12;
        // Specify your bucket Region (an example Region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            string urlString = GeneratePreSignedURL(timeoutDuration);
        }
        static string GeneratePreSignedURL(double duration)
        {
            string urlString = "";
            try
            {
                GetPreSignedUrlRequest request1 = new GetPreSignedUrlRequest
                {
                    BucketName = accessPointArn,
                    Key = objectKey,
                    Expires = DateTime.UtcNow.AddHours(duration)
                };
                urlString = s3Client.GetPreSignedURL(request1);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            return urlString;
        }
```

```
    }
}
```

Python

下列範例使用了 SDK for Python (Boto3) 來產生預先簽章的 URL 以共用物件。例如，使用 Boto3 用戶端和 `generate_presigned_url` 函數來產生允許您 `GET` 物件的預先簽章的 URL。

```
import boto3
    url = boto3.client('s3').generate_presigned_url(
    ClientMethod='get_object',
    Params={'Bucket': 'ACCESS_POINT_ARN', 'Key': 'OBJECT_KEY'},
    ExpiresIn=3600)
```

如需有關使用 SDK for Python (Boto3) 產生預先簽章的 URL 的詳細資訊，請參閱《AWS SDK for Python (Boto) API 參考》中的「Python」。

使用 AWS CLI

下列範例 AWS CLI 命令會為 S3 on Outposts 儲存貯體產生預先簽章的 URL。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

> **Note**
>
> 當您使用 AWS CLI 產生預先簽章的 URL 時，預先簽章 URL 的過期時間上限為建立時間起算 7 天。

```
aws s3 presign s3://arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-
point/mydoc.txt --expires-in 604800
```

如需詳細資訊，請參閱《AWS CLI 命令參考》中的 presign。

產生預先簽章 URL 以將物件上傳至 S3 on Outposts 儲存貯體

若要授予對存放在本機 Outpost 上物件的有限時間存取權限，而不會更新儲存貯體政策，您可以使用預先簽章 URL。使用預先簽章 URL，身為儲存貯體擁有者的您可以與虛擬私有雲端 (VPC) 中的個人共享物件，或授予他們上傳或刪除物件的能力。

當您使用 AWS SDKs或 AWS Command Line Interface (AWS CLI) 建立預先簽章的 URL 時，您可以將 URL 與特定動作建立關聯。您也可以選擇自訂到期時間 (最低 1 秒，最高 7 天) 來授予預先簽章 URL 有限時間的存取權。當您共用預先簽章 URL 時，VPC 中的個人可以執行內嵌在 URL 中的動作，如同原始簽章使用者一樣。當 URL 到達到期時間時，該 URL 就會過期且再也無法運作。

當您建立預先簽章 URL 時，必須提供安全憑證，然後指定下列項目：

- 適用於 Amazon S3 on Outposts 儲存貯體的存取點 Amazon Resource Name (ARN)
- 物件索引鍵
- HTTP 方法 (PUT 用於上傳物件)
- 過期日期和時間

預先簽章 URL 僅在指定的期間內有效。也就是說，您必須在到期日期和時間之前開始 URL 所允許的操作。您可以多次使用預先簽章 URL，直到到期日期和時間為止。如果使用暫時字符建立了預先簽章的 URL，那麼 URL 會在字符過期時過期，即使您使用較晚的過期時間建立 URL 亦然。

如果預先簽章的 URL 所允許的動作包含多個步驟 (例如分段上傳)，則您必須在到期之前開始所有步驟。如果 S3 on Outposts 嘗試以過期 URL 開始步驟時，您會收到錯誤。

虛擬私有雲端 (VPC) 中可存取預先簽章 URL 的使用者可以上傳物件。例如，VPC 中具有可存取預先簽章 URL 的使用者可以將物件上傳到您的儲存貯體。由於預先簽章 URL 會將 S3 on Outposts 儲存貯體的存取權授予 VPC 中擁有預先簽章 URL 存取權的任何使用者，因此我們建議您妥善保護這些 URL。如需有關保護預先簽署 URL 的詳細資訊，請參閱限制預先簽章的 URL 功能。

任何具備有效安全憑證的使用者，均可建立預先簽章的 URL。然而，只有具備許可執行作為預先簽章 URL 基礎操作的人員，才能建立預先簽章 URL。如需詳細資訊，請參閱誰可以建立預先簽章的 URL。

使用 AWS SDKs為 S3 on Outposts 物件操作產生預先簽章的 URL

Java

SDK for Java 2.x

此範例顯示如何產生可以於限定時間內用來將物件上傳至 S3 on Outposts 儲存貯體的預先簽章 URL。如需詳細資訊，請參閱使用適用於 S3 on OutOutposts 的預先簽章 URL。

```
   public static void signBucket(S3Presigner presigner, String
outpostAccessPointArn, String keyName) {

       try {
```

```
            PutObjectRequest objectRequest = PutObjectRequest.builder()
                    .bucket(accessPointArn)
                    .key(keyName)
                    .contentType("text/plain")
                    .build();

            PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
                    .signatureDuration(Duration.ofMinutes(10))
                    .putObjectRequest(objectRequest)
                    .build();

            PresignedPutObjectRequest presignedRequest =
presigner.presignPutObject(presignRequest);


            String myURL = presignedRequest.url().toString();
            System.out.println("Presigned URL to upload a file to: " +myURL);
            System.out.println("Which HTTP method must be used when uploading a
file: " +
                    presignedRequest.httpRequest().method());

            // Upload content to the S3 on Outposts bucket by using this URL.
            URL url = presignedRequest.url();

            // Create the connection and use it to upload the new object by using
the presigned URL.
            HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
            connection.setDoOutput(true);
            connection.setRequestProperty("Content-Type","text/plain");
            connection.setRequestMethod("PUT");
            OutputStreamWriter out = new
OutputStreamWriter(connection.getOutputStream());
            out.write("This text was uploaded as an object by using a presigned
URL.");
            out.close();

            connection.getResponseCode();
            System.out.println("HTTP response code is " +
connection.getResponseCode());

        } catch (S3Exception e) {
            e.getStackTrace();
```

```
        } catch (IOException e) {
            e.getStackTrace();
        }
    }
```

Python

SDK for Python (Boto3)

此範例顯示如何產生可於限定時間內執行 S3 on Outposts 動作的預先簽章 URL。如需詳細資訊，請參閱使用適用於 S3 on OutOutposts 的預先簽章 URL。若要使用 URL 提出請求，請使用 Requests 套件。

```
import argparse
import logging
import boto3
from botocore.exceptions import ClientError
import requests

logger = logging.getLogger(__name__)


def generate_presigned_url(s3_client, client_method, method_parameters,
 expires_in):
    """
    Generate a presigned S3 on Outposts URL that can be used to perform an
 action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds that the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method,
            Params=method_parameters,
            ExpiresIn=expires_in
        )
        logger.info("Got presigned URL: %s", url)
    except ClientError:
```

```
        logger.exception(
            "Couldn't get a presigned URL for client method '%s'.",
 client_method)
        raise
    return url


def usage_demo():
    logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')

    print('-'*88)
    print("Welcome to the Amazon S3 on Outposts presigned URL demo.")
    print('-'*88)

    parser = argparse.ArgumentParser()
    parser.add_argument('accessPointArn', help="The name of the S3 on Outposts
 access point ARN.")
    parser.add_argument(
        'key', help="For a GET operation, the key of the object in S3 on
 Outposts. For a "
                    "PUT operation, the name of a file to upload.")
    parser.add_argument(
        'action', choices=('get', 'put'), help="The action to perform.")
    args = parser.parse_args()

    s3_client = boto3.client('s3')
    client_action = 'get_object' if args.action == 'get' else 'put_object'
    url = generate_presigned_url(
        s3_client, client_action, {'Bucket': args.accessPointArn, 'Key':
 args.key}, 1000)

    print("Using the Requests package to send a request to the URL.")
    response = None
    if args.action == 'get':
        response = requests.get(url)
    elif args.action == 'put':
        print("Putting data to the URL.")
        try:
            with open(args.key, 'r') as object_file:
                object_text = object_file.read()
            response = requests.put(url, data=object_text)
        except FileNotFoundError:
            print(f"Couldn't find {args.key}. For a PUT operation, the key must
 be the "
```

```
                    f"name of a file that exists on your computer.")

    if response is not None:
        print("Got response:")
        print(f"Status: {response.status_code}")
        print(response.text)

    print('-'*88)


if __name__ == '__main__':
    usage_demo()
```

搭配使用 Amazon S3 on Outposts 和本機 Amazon EMR on Outposts

Amazon EMR 是一種受管叢集平台，可簡化在 上執行大數據架構，例如 Apache Hadoop和 Apache Spark， AWS 以處理和分析大量資料。透過使用這些架構和相關的開放原始碼專案，您可以處理用於分析用途和商業智慧工作負載的資料。Amazon EMR 也可協助您轉換大量資料，並將其移入和移出其他 AWS 資料存放區和資料庫，並支援 Amazon S3 on Outposts。如需 Amazon EMR 的詳細資訊，請參閱《Amazon EMR 管理指南》中的 Amazon EMR on Outposts。

對於 Amazon S3 on Outposts，Amazon EMR 7.0.0 版已支援的 Apache Hadoop S3A 連接器。Amazon EMR 的早期版本不支援本機 S3 on Outposts，且不支援 EMR 檔案系統 (EMRFS)。

支援的應用程式

搭配 Amazon S3 on Outposts 的 Amazon EMR 支援下列應用程式：

- Hadoop
- Spark
- Hue
- Hive
- Sqoop
- Pig
- Hudi
- Flink

如需詳細資訊，請參閱《Amazon EMR 版本指南》。

建立和設定 Amazon S3 on Outposts 儲存貯體

Amazon EMR 使用 適用於 Java 的 AWS SDK 搭配 Amazon S3 on Outposts 來存放輸入資料和輸出資料。您的 Amazon EMR 日誌檔案會儲存在您選擇的區域 Amazon S3 位置，而不會以本機方式儲存在 Outpost 上。如需詳細資訊，請參閱《Amazon EMR 管理指南》中的 Amazon EMR 日誌。

為了符合 Amazon S3 和 DNS 需求，S3 on Outposts 儲存貯體具有特定的命名約束與限制。如需詳細資訊，請參閱建立 S3 on Outposts 儲存貯體。

使用 Amazon EMR 7.0.0 版及更新版本時，您可以將 Amazon EMR 與 S3 on Outposts 和 S3A 檔案系統搭配使用。

先決條件

S3 on Outposts 許可 – 當您建立 Amazon EMR 執行個體描述檔時，您的角色必須包含 S3 on Outposts 的 AWS Identity and Access Management (IAM) 命名空間。S3 on Outposts 具有自己的命名空間 `s3-outposts*`。如需有關使用此命名空間的政策範例，請參閱使用 S3 on Outposts 設定 IAM。

S3A 連接器 – 若要設定 EMR 叢集，以從 Amazon S3 on Outposts 儲存貯體存取資料，您必須使用 Apache Hadoop S3A 連接器。若要使用該連接器，請確保您的所有 S3 URI 均使用 `s3a` 結構描述。如果未使用，您可以設定用於 EMR 叢集的檔案系統實作，以便搭配使用您的 S3 URI 與 S3A 連接器。

若要將檔案系統實作設定為使用 S3A 連接器，您可以使用 EMR 叢集的 `fs.`*`file_scheme`*`.impl` 和 `fs.AbstractFileSystem.`*`file_scheme`*`.impl` 組態屬性，其中 *`file_scheme`* 對應於您擁有的 S3 URI 類型。若要使用下列範例，請以您自己的資訊取代 *`user input placeholders`*。例如，若要變更使用 `s3` 結構描述之 S3 URI 的檔案系統實作，請指定下列叢集組態屬性：

```
[
  {
"Classification": "core-site",
    "Properties": {
    "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
    "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

若要使用 S3A，請將 `fs.file_scheme.impl` 組態屬性設定為 `org.apache.hadoop.fs.s3a.S3AFileSystem`，並將 `fs.AbstractFileSystem.file_scheme.impl` 屬性設定為 `org.apache.hadoop.fs.s3a.S3A`。

例如，如果您存取路徑 `s3a://bucket/...`，請將 `fs.s3a.impl` 屬性設定為 `org.apache.hadoop.fs.s3a.S3AFileSystem`，並將 `fs.AbstractFileSystem.s3a.impl` 屬性設定為 `org.apache.hadoop.fs.s3a.S3A`。

搭配使用 Amazon S3 on Outposts 和 Amazon EMR 的入門指引

下列主題說明如何開始使用 Amazon EMR 搭配 Amazon S3 on Outposts。

主題

建立許可政策

在建立使用 Amazon S3 on Outposts 的 EMR 叢集之前，您必須先建立 IAM 政策以連接至該叢集的 Amazon EC2 執行個體設定檔。政策必須具有存取 S3 on Outposts 存取點 Amazon Resource Name (ARN) 的許可。如需有關為 S3 on Outposts 建立 IAM 政策的詳細資訊，請參閱使用 S3 on Outposts 設定 IAM。

下列政策範例會示範如何授予所需的許可。建立政策後，將政策連接至用於建立 EMR 叢集的執行個體設定檔角色，如 the section called “建立和設定叢集” 一節中所述。若要使用此範例，請以您自己的資訊取代 *user input placeholders*。

```
{
"Version":"2012-10-17",
  "Statement": [
        {
  "Effect": "Allow",
            "Resource": "arn:aws:s3-outposts:us-
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name",
            "Action": [
                "s3-outposts:*"
```

```
            ]
        }
    ]

}
```

建立和設定叢集

若要建立使用 S3 on Outposts 執行 Spark 的叢集，請在主控台中完成下列步驟。

建立使用 S3 on Outposts 執行 Spark 的叢集

1. 請在 https://console.aws.amazon.com/elasticmapreduce/ 開啟 Amazon EMR 主控台。
2. 在左側導覽窗格中選擇叢集。
3. 選擇 建立叢集。
4. 針對 Amazon EMR 版本，請選擇 emr-7.0.0 或更新版本。
5. 針對應用程式套件，請選擇 Spark 互動式。然後選取您要包含在叢集上的任何其他支援應用程式。
6. 若要啟用 Amazon S3 on Outposts，請輸入您的組態設定。

   範例組態設定

   若要使用下列範例組態設定，請以您自己的資訊取代 *user input placeholders*。

```
[
 {
   "Classification": "core-site",
   "Properties": {
     "fs.s3a.bucket.DOC-EXAMPLE-BUCKET.accesspoint.arn": "arn:aws:s3-outposts:us-
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name"
     "fs.s3a.committer.name": "magic",
     "fs.s3a.select.enabled": "false"
    }
  },
  {
    "Classification": "hadoop-env",
    "Configurations": [
      {
        "Classification": "export",
        "Properties": {
```

```
          "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
          }
       }
     ],
     "Properties": {}
   },
   {
     "Classification": "spark-env",
     "Configurations": [
       {
         "Classification": "export",
         "Properties": {
           "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
         }
       }
      ],
      "Properties": {}
     },
     {
      "Classification": "spark-defaults",
      "Properties": {
        "spark.executorEnv.JAVA_HOME": "/usr/lib/jvm/java-11-amazon-
corretto.x86_64",
        "spark.sql.sources.fastS3PartitionDiscovery.enabled": "false"
      }
     }
  ]
```

7. 在聯網區段中，選擇 AWS Outposts 機架上的虛擬私有雲端 (VPC) 和子網路。如需 Outposts 上的 Amazon EMR 的詳細資訊，請參閱《Amazon EMR 管理指南》內之 AWS Outposts上的 EMR 叢集。
8. 在適用於 Amazon EMR 的 EC2 執行個體設定檔區段中，選擇連接您先前建立之許可政策的 IAM 角色。
9. 設定剩餘的叢集設定，然後選擇建立叢集。

組態概觀

下表會說明 S3A 組態，以及當您設定使用 S3 on Outposts 搭配 Amazon EMR 的叢集時，要為其參數指定的值。

參數	預設值	S3 on Outposts 的必要值	說明
fs.s3a.aws.credentials.provider	如果未指定，S3A 會尋找區域儲存貯體 (具有 Outposts 儲存貯體名稱) 中的 S3。	S3 on Outposts 儲存貯體的存取點 ARN	Amazon S3 on Outposts 支援僅限虛擬私有雲端 (VPC) 的存取點來作為存取 Outpost 儲存貯體的唯一方法。
fs.s3a.committer.name	file	magic	Magic Committer 是 S3 on Outposts 唯一支援的遞交程式。
fs.s3a.select.enabled	TRUE	FALSE	Outposts 上不支援 S3 Select。
JAVA_HOME	/usr/lib/jvm/java-8	/usr/lib/jvm/java-11-amazon-corretto.x86_64	S3A 上的 S3 on Outposts 需要 Java 11 版。

下表說明當您設定搭配 Amazon EMR 使用 S3 on Outposts 的叢集時，要為其參數指定的 Spark 組態和值。

參數	預設值	S3 on Outposts 的必要值	說明
spark.sql.sources.fastS3PartitionDiscovery.enabled	TRUE	FALSE	S3 on Outposts 不支援快速分割區。
spark.executorEnv.JAVA_HOME	/usr/lib/jvm/java-8	/usr/lib/jvm/java-11-amazon	S3A 上的 S3 on Outposts 需要 Java 11 版。

參數	預設值	S3 on Outposts 的必要值	說明
		`-corretto.x86_64`	

考量事項

當您將 Amazon EMR 與 S3 on Outposts 儲存貯體整合時，請考慮下列事項：

- Amazon EMR 7.0.0 版及更新版本支援 Amazon S3 on Outposts。
- 您需要 S3A 連接器才能將 S3 on Outposts 與 Amazon EMR 搭配使用。只有 S3A 具有與 S3 on Outposts 儲存貯體互動所需的功能。如需 S3A 連接器設定資訊，請參閱先決條件。
- Amazon S3 on Outposts 僅支援使用 Amazon S3 受管金鑰 (SSE-S3) 搭配 Amazon EMR 的伺服器端加密。如需詳細資訊，請參閱the section called “資料加密”。
- Amazon S3 on Outposts 不支援使用 S3A FileOutputCommitter 進行寫入。使用 S3 on Outposts 儲存貯體上的 S3A FileOutputCommitter 寫入會導致下列錯誤：InvalidStorageClass: The storage class you specified is not valid。
- Amazon EMR Serverless 或 Amazon EMR on EKS 不支援 Amazon S3 on Outposts。
- Amazon EMR 日誌會儲存在您選擇的區域 Amazon S3 位置，而不會以本機方式儲存在 S3 on Outposts 儲存貯體中。

授權與身分驗證快取

S3 on Outposts 會安全地快取 Outposts 機架上的本機身分驗證和授權資料。快取會移除 AWS 區域每個請求的父系往返。這可消除網路往返時造成的變異。透過 S3 on Outposts 中的身分驗證和授權快取，您可以體驗一致延遲表現 (不同與 Outposts 和 AWS 區域之間的連線延遲)。

當您提出 S3 on Outposts API 請求時，系統會安全地快取身分驗證和授權資料。然後，系統會使用快取資料來驗證後續的 S3 物件 API 請求。S3 on Outposts 只會在使用簽章版本 4A (SigV4A) 簽署請求時，快取身分驗證和授權資料。快取會以本機方式儲存在 S3 on Outposts 服務內的 Outposts 上。當您提出 S3 API 請求時，系統會以非同步方式重新整理快取。系統會將快取加密，而且不會將純文字密碼編譯金鑰儲存在 Outposts 上。

當 Outpost 連線到 AWS 區域時，快取的有效期最長為 10 分鐘。當您提出 S3 on Outposts API 請求時，系統會以非同步方式重新整理快取，以確保使用最新的政策。如果 Outpost 與 中斷連線 AWS 區域，快取的有效期最長為 12 小時。

設定授權和身分驗證快取

S3 on Outposts 會自動快取使用 SigV4A 演算法簽署之請求的身分驗證和授權資料。如需詳細資訊，請參閱AWS Identity and Access Management 《 使用者指南》中的簽署 AWS API 請求。SigV4A 演算法可在最新版本 AWS SDKs 中使用。您可以透過 AWS Common Runtime (CRT) 程式庫的相依性來取得該項目。

您需要使用最新版本的 AWS SDK，並安裝最新版本的 CRT。例如，您可以執行 `pip install awscrt` 來取得最新版本的 CRT (具有 Boto3)。

S3 on Outposts 不會快取使用 SigV4 演算法簽署之請求的身分驗證和授權資料。

驗證 SigV4A 簽署

您可以使用 AWS CloudTrail 驗證請求是否已使用 SigV4A 簽署。如需為 S3 on Outposts 設定 CloudTrail 的詳細資訊，請參閱使用 AWS CloudTrail 日誌監控 S3 on Outposts。

設定 CloudTrail 之後，您可以在 CloudTrail 日誌的 `SignatureVersion` 欄位中驗證請求的簽署方式。使用 SigV4A 簽署的請求會將 `SignatureVersion` 設為 `AWS 4-ECDSA-P256-SHA256`。使用 SigV4 簽署的請求會將 `SignatureVersion` 設為 `AWS 4-HMAC-SHA256`。

S3 on Outposts 中的安全性

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是專為滿足最安全敏感組織的需求而建置。

安全是 AWS 與您之間的共同責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 AWS 服務 中執行的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。在AWS 合規計畫中，第三方稽核人員會定期測試和驗證我們安全的有效性若要了解適用於 Amazon S3 on Outposts 的合規計劃，請參閱AWS 合規計劃的 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 服務 的 。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件有助於您了解如何在使用 S3 on Outposts 時套用共同的責任模型。下列主題說明如何將 S3 on Outposts 設定為達到您的安全及合規目標。您也會了解如何使用其他 AWS 服務 來協助您監控和保護 S3 on Outposts 資源。

主題

- 使用 S3 on Outposts 設定 IAM
- S3 on Outposts 中的資料加密
- AWS PrivateLink 適用於 S3 on Outposts
- AWS Signature 第 4 版 (SigV4) 身分驗證特定政策金鑰
- AWS Amazon S3 on Outposts 的 受管政策
- 針對 Amazon S3 on Outposts 使用服務連結角色

使用 S3 on Outposts 設定 IAM

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制哪些人員可進行身分驗證 (登入) 並獲得授權 (具有許可) 以使用 Amazon S3 on Outposts 資源。IAM 是一種您可以免費使用的 AWS 服務 。根據預設，使用者不具備 S3 on Outposts 資源和操作的許可。若要授予 S3 on Outposts 資源和 API 操作的存取權限，您可以使用 IAM 建立使用者、群組或角色，並附加許可。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。

- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。遵循「IAM 使用者指南」的為第三方身分提供者 (聯合) 建立角色中的指示。

- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照「IAM 使用者指南」的為 IAM 使用者建立角色中的指示。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台) 中的指示。

除了 IAM 身分型政策外，S3 on Outposts 也同時支援儲存貯體和存取點政策。儲存貯體政策與存取點政策是連接到 S3 on Outposts 資源的資源型政策。

- 儲存貯體政策連接到儲存貯體，並根據政策中的元素來允許或拒絕對儲存貯體和其中物件的請求。
- 相比之下，存取點原則連接到存取點，並允許或拒絕對存取點的要求。

存取點政策可搭配連接到基礎 S3 on Outposts 儲存貯體的儲存貯體政策。若要讓應用程式或使用者能夠透過 S3 on Outposts 存取點來存取 S3 on Outposts 儲存貯體中的物件，則存取點政策和儲存貯體政策皆必須允許該請求。

您在存取點政策中包含的限制僅適用透過該存取點進行的請求。例如，如果存取點連接到儲存貯體，則無法使用存取點政策來允許或拒絕直接向儲存貯體提出的請求。不過，您套用至儲存貯體政策的限制可能會允許或拒絕直接向儲存貯體或透過存取點提出的請求。

在 IAM 政策或資源型政策中，您可以定義哪些 S3 on Outposts 動作會受到允許或拒絕。S3 on Outposts 動作對應特定的 S3 on Outposts API 操作。S3 on Outposts 動作會使用 `s3-outposts:` 命名空間字首。對 中的 S3 on Outposts 控制 API 提出的請求， AWS 區域 以及對 Outpost 上的物件 API 端點提出的請求，都會使用 IAM 進行驗證，並根據`s3-outposts:`命名空間字首進行授權。若要使用 S3 on Outposts，請設定您的 IAM 使用者，並針對 `s3-outposts:` IAM 命名空間授權使用者。

如需詳細資訊，請參閱《服務授權參考》中的「S3 on Outposts 的動作、資源和條件金鑰」。

> **Note**
>
> - S3 on Outposts 不支援存取控制清單 (ACL)。
> - Posts 上的 S3 預設會將儲存貯體擁有者做為物件擁有者，以協助確保儲存貯體的擁有者無法存取或刪除物件。

- Outpost 上的 S3 一律會啟用 S3 封鎖公有存取權限，以協助確保物件永遠不會有公有存取權限。

如需有關 S3 on Outposts 設定 IAM 的詳細資訊，請參下列主題。

主題

適用於 S3 on Outposts 政策的主體

當您建立資源型政策以授予 S3 on Outposts 儲存貯體的存取權時，您必須使用 `Principal` 元素來指定可對該資源提出動作或操作請求的人員或應用程式。對於 S3 on Outposts 政策，您可以使用下列其中一項主體：

- 一個 AWS 帳戶
- IAM 使用者
- IAM 角色
- 政策中使用 `Condition` 元素來限制對特定 IP 範圍存取的所有主體 (指定萬用字元 *)

> **Important**
>
> 您無法針對在 `Principal` 元素中使用萬用字元 (*) 的 S3 on Outposts 儲存貯體來撰寫政策，除非該政策還包含限制對特定 IP 地址範圍存取的 `Condition`。此限制協助確保無法公有存取 S3 on Outposts 儲存貯體。如需範例，請參閱「適用於 S3 on Outposts 的範例政策」。

如需有關 `Principal` 元素的詳細資訊，請參閱《IAM 使用者指南》中的「AWS JSON 政策元素：主體」。

適用於 S3 on Outposts 的資源 ARN

S3 on Outposts 的 Amazon Resource Name (ARNs) 包含 Outpost ID，以及 AWS 區域 Outpost 所在的 、 AWS 帳戶 ID 和資源名稱。若要存取並對 Outposts 儲存貯體和物件執行動作，您必須使用下表中顯示的其中一個 ARN 格式。

ARN 中的*partition*值是指 的群組 AWS 區域。每個 的範圍 AWS 帳戶 都是一個分割區。以下是支援的分割區：

- `aws` – AWS 區域
- `aws-us-gov` – AWS GovCloud (US) 區域

下表顯示 S3 on Outposts ARN 格式。

Amazon S3 on Outposts ARN	ARN 格式	範例
儲存貯體 ARN	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /bucket/*bucket_name*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /bucket/*amzn-s3-demo-bucket1*
存取點 ARN	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /accesspoint/ *accesspoint_name*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /accesspoint/ *access-point-name*
物件 ARN	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /

Amazon S3 on Outposts ARN	ARN 格式	範例
	bucket/*bucket_name* / object/*object_key*	bucket/*amzn-s3-demo-bucket1* /object/*myobject*
S3 on Outposts 存取點物件 ARN (用於原則)	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /accesspoint/ *accesspoint_name* / object/*object_key*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* : outpost/ *op-01ac5d28a6a232904* /accesspoint/ *access-point-name/object/myobject*
S3 on Outposts ARN	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* : outpost/ *op-01ac5d28a6a232904*

適用於 S3 on Outposts 的範例政策

Example ：具有 AWS 帳戶 委託人的 S3 on Outposts 儲存貯體政策

下列儲存貯體政策使用 AWS 帳戶 委託人來授予對 S3 on Outposts 儲存貯體的存取權。若要使用此儲存貯體政策，請以您自己的資訊取代 *user input placeholders*。

```
{
   "Version":"2012-10-17",
   "Id":"ExampleBucketPolicy1",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Principal":{
            "AWS":"123456789012"
         },
         "Action":"s3-outposts:*",
```

```
         "Resource":"arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket"
      }
   ]
}
```

Example ：使用萬用字元 (*) 主體和條件金鑰的 S3 on Outposts 儲存貯體政策，以限制對特定 IP 地址範圍的存取

下列儲存貯體政策使用萬用字元主體 (*) 搭配 aws:SourceIp 條件以限制對特定 IP 地址範圍的存取。若要使用此儲存貯體政策，請以您自己的資訊取代 *user input placeholders*。

```
{
    "Version": "2012-10-17",
    "Id": "ExampleBucketPolicy2",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": { "AWS" : "*" },
            "Action":"s3-outposts:*",
            "Resource":"arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket",
            "Condition" : {
                "IpAddress" : {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NotIpAddress" : {
                    "aws:SourceIp": "198.51.100.0/24"
                }
            }
        }
    ]
}
```

適用於 S3 on Outposts 端點的許可

S3 on Outposts 需要 IAM 中擁有的許可，來管理 S3 on Outposts 端點動作。

> **Note**
>
> - 對於使用客戶擁有的 IP 地址集區 (CoIP 集區) 存取類型的端點，您也必須具有從 CoIP 集區使 IP 地址的許可，如下表所述。
> - 對於使用 存取 S3 on Outposts 的共用帳戶 AWS Resource Access Manager，這些共用帳戶中的使用者無法在共用子網路上建立自己的端點。如果共用帳戶中的使用者想要管理自己的端點，則共用帳戶必須在 Outpost 上建立自己的子網路。如需詳細資訊，請參閱the section called “共用 S3 on Outposts”。

下表顯示與 S3 on Outposts 端點相關的 IAM 許可。

動作	IAM 許可

`CreateEndpoint`	`s3-outposts:CreateEndpoint`
	
	`ec2:CreateNetworkInterface`
	
	`ec2:DescribeNetworkInterfaces`
	
	`ec2:DescribeVpcs`
	
	`ec2:DescribeSecurityGroups`
	
	`ec2:DescribeSubnets`
	
	`ec2:CreateTags`
	
	`iam:CreateServiceLinkedRole`
	
	對於使用內部部署客戶擁有的 IP 地址集區 (CoIP 集區) 存取類型的端點，需要下列額外的許可：
	
	`s3-outposts:CreateEndpoint`
	
	`ec2:DescribeCoipPools`
	
	`ec2:GetCoipPoolUsage`

動作	IAM 許可
	ec2:AllocateAddress ec2:AssociateAddress ec2:DescribeAddresses ec2:DescribeLocalGatewayRouteTableVpcAssociations
DeleteEndpoint	s3-outposts:DeleteEndpoint ec2:DeleteNetworkInterface ec2:DescribeNetworkInterfaces 對於使用內部部署客戶擁有的 IP 地址集區 (CoIP 集區) 存取類型的端點，需要下列額外的許可： s3-outposts:DeleteEndpoint ec2:DisassociateAddress ec2:DescribeAddresses ec2:ReleaseAddress
ListEndpoints	s3-outposts:ListEndpoints

> **Note**
>
> 您可以在 IAM 政策中使用資源標籤來管理許可。

S3 on Outposts 的服務連結角色

S3 on Outposts 使用 IAM 服務連結角色代表您建立一些網路資源。如需詳細資訊，請參閱[針對 Amazon S3 on Outposts 使用服務連結角色]。

S3 on Outposts 中的資料加密

依預設，所有存放在 Amazon S3 on Outposts 中的資料均會使用伺服器端加密與 Amazon S3 受管加密金鑰 (SSE-S3) 進行加密。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的搭配 Amazon S3 受管金鑰 (SSE-S3) 使用伺服器端加密。

您可以選擇性以客戶提供的加密金鑰 (SSE-C) 使用伺服器端加密。若要使用 SSE-C，請指定加密金鑰做為物件 API 請求的一部分。伺服器端加密只會加密物件資料，非物件中繼資料。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的搭配客戶提供的金鑰使用伺服器端加密。

> **Note**
>
> S3 on Outposts 不支援使用 AWS Key Management Service (AWS KMS) 金鑰 (SSE-KMS) 的伺服器端加密。

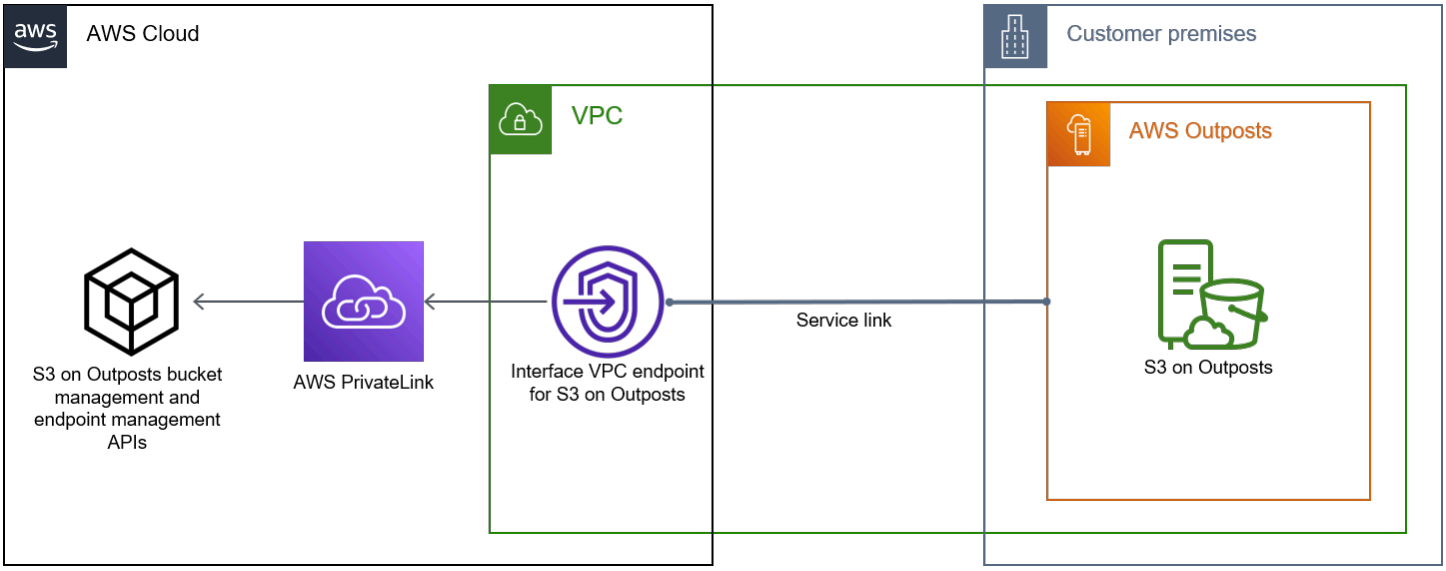
AWS PrivateLink 適用於 S3 on Outposts

S3 on Outposts 支援 AWS PrivateLink，可讓您透過虛擬私有網路中的私有端點，直接管理對 S3 on Outposts 儲存體的存取。如此您就能使用自己的虛擬私有雲端 (VPC) 中的私有 IP 地址，進而簡化內部網路架構並在您的 Outposts 物件儲存上執行管理操作。使用 AWS PrivateLink 不需要使用公有 IP 地址或代理伺服器。

使用 AWS PrivateLink for Amazon S3 on Outposts，您可以在虛擬私有雲端 (VPC) 中佈建介面 VPC 端點，以存取 S3 on Outposts 儲存貯體管理和端點管理 APIs。您可透過虛擬私有網路 (VPN) 或 AWS Direct Connect，直接從部署在 VPC 中或內部部署中的應用程式存取介面 VPC 端點。您可以透過 存取儲存貯 AWS PrivateLink 體和端點管理 APIs AWS PrivateLink。不支援資料傳輸 API 操作，例如 GET、PUT 和類似的 APIs。這些操作已透過 S3 on Outposts 端點和存取點組態私下傳輸。如需詳細資訊，請參閱適用於 S3 on Outposts 的網路。

介面端點由一個或多個彈性網路介面 (ENI) 來表示，這些是在 VPC 的子網路中指派的私有 IP 地址。對 S3 on Outposts 介面端點發出的請求會自動路由至 AWS 網路上的 S3 on Outposts 儲存貯體和端點管理 API。您也可以透過 AWS Direct Connect 或 AWS Virtual Private Network ()，從內部部署應用程式存取 VPC 中的界面端點AWS VPN。如需有關如何將 VPC 與內部部署網路連線的詳細資訊，請參閱《AWS Direct Connect 使用者指南》和《AWS Site-to-Site VPN 使用者指南》。

介面端點會透過 AWS 網路和透過 路由 S3 on Outposts 儲存貯體和端點管理 APIs請求 AWS PrivateLink，如下圖所示。



如需有關介面端點的一般資訊，請參閱《AWS PrivateLink指南》中的介面 VPC 端點 (AWS PrivateLink)。

主題

- 法規與限制
- 存取 S3 on Outposts 介面端點
- 更新內部部署 DNS 組態
- 為 S3 on Outposts 建立 VPC 端點政策
- 為 S3 on Outposts 建立儲存貯體政策與 VPC 端點政策

法規與限制

當您透過 存取 S3 on Outposts 儲存貯體和端點管理 APIs 時 AWS PrivateLink，會套用 VPC 限制。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的介面端點屬性和限制以及AWS PrivateLink 配額。

此外， AWS PrivateLink 不支援下列項目：

- 聯邦資訊處理標準 (FIPS) 端點
- S3 on Outposts 資料傳輸 API，例如 GET、PUT 和類似的物件 API 操作。
- 私有 DNS

存取 S3 on Outposts 介面端點

若要使用 存取 S3 on Outposts 儲存貯體和端點管理 APIs AWS PrivateLink，您必須更新您的應用程式以使用端點特定的 DNS 名稱。當您建立介面端點時， AWS PrivateLink 會產生兩種端點特定的 S3 on Outposts 名稱：區域和區域。

- 區域 DNS 名稱 – 包含唯一的 VPC 端點 ID、服務識別符 AWS 區域、 和 `vpce.amazonaws.com`，例如 *`vpce-1a2b3c4d-5e6f`*`.s3-outposts.us-east-1.vpce.amazonaws.com`。
- 區域 DNS 名稱 – 包含唯一的 VPC 端點 ID、可用區域、服務識別符 AWS 區域、 和 `vpce.amazonaws.com`，例如 *`vpce-1a2b3c4d-5e6f`*`-us-east-1a.s3-outposts.us-east-1.vpce.amazonaws.com`。如果您的架構可隔離可用區域，則可以使用此選項。例如，您可以將地區 DNS 名稱用於故障遏止或降低區域資料傳輸成本。

> **Important**
>
> S3 on Outposts 介面端點是從公有 DNS 網域解析而得。S3 on Outposts 不支援私有 DNS。針對所有儲存貯體與端點管理 API 使用 `--endpoint-url` 參數。

AWS CLI 範例

使用 `--region` 和 `--endpoint-url` 參數透過 S3 on Outposts 介面端點存取儲存貯體管理與端點管理 API。

Example ：使用端點 URL 列出具有 S3 控制項 API 的儲存貯體

在下列範例中，將區域 *`us-east-1`*、VPC 端點 URL *`vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`* 及帳戶 ID *`111122223333`* 取代為適當的資訊。

```
aws s3control list-regional-buckets --region us-east-1 --endpoint-url
 https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com --account-
id 111122223333
```

AWS SDK 範例

將 SDK 更新至最新版本，並設定您的用戶端使用端點 URL，以存取 S3 on Outposts 介面端點的 S3 控制 API。

SDK for Python (Boto3)

Example ：使用端點 URL 存取 S3 控制 API

在下列範例中，將區域 *us-east-1* 和 VPC 端點 URL *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* 取代為適當的資訊。

```
control_client = session.client(
service_name='s3control',
region_name='us-east-1',
endpoint_url='https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com'
)
```

如需詳細資訊，請參閱《Boto3 開發人員指南》中的 AWS PrivateLink for Amazon S3。

SDK for Java 2.x

Example ：使用端點 URL 存取 S3 控制 API

在下列範例中，將 VPC 端點 URL *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* 和區域 *Region.US_EAST_1* 取代為適當的資訊。

```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

 .endpointOverride(URI.create("https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com"))
                                  .build()
```

如需詳細資訊，請參閱 適用於 Java 的 AWS SDK API 參考中的 S3ControlClient。

更新內部部署 DNS 組態

使用端點特定 DNS 名稱來存取 S3 on Outposts 儲存貯體管理與端點管理 API 的介面端點時，您不必更新內部部署 DNS 解析器。您可以使用公有 S3 on Outposts DNS 網域中介面端點的私有 IP 地址，來解析端點特定的 DNS 名稱。

為 S3 on Outposts 建立 VPC 端點政策

若要為 S3 on Outposts 建立 VPC 介面端點，請參閱 AWS PrivateLink 指南中的建立 VPC 端點。

為 S3 on Outposts 建立儲存貯體政策與 VPC 端點政策

您可以將端點政策連接至控制 S3 on Outposts 存取權的 VPC 端點。您還可以使用 S3 on Outposts 儲存貯體政策的 `aws:sourceVpce` 條件，來限制特定 VPC 端點對特定儲存貯體的存取。透過 VPC 端點政策，您可以控制存取 S3 on Outposts 儲存貯體管理 API 與端點管理 API。透過儲存貯體政策，您可以控制存取 S3 on Outposts 儲存貯體管理 API。然而，您無法使用 `aws:sourceVpce` 管理對 S3 on Outposts 其物件動作的存取。

S3 on Outposts 的存取政策指定下列資訊：

- 允許或拒絕動作的 AWS Identity and Access Management (IAM) 主體。
- 遭允許或拒絕的 S3 控制項動作。
- 遭允許或拒絕其動作的 S3 on Outposts 資源。

下列範例顯示了限制儲存貯體或端點存取權的政策。如需 VPC 連線的詳細資訊，請參閱 AWS Network-to-VPC 連線選項 Amazon Virtual Private Cloud。

> **Important**
>
> - 當套用本節所述的 VPC 端點範例政策時，您可能會在無意間封鎖您對儲存貯體的存取。會限制儲存貯體存取源自您 VPC 端點之連線的儲存貯體許可，可能會封鎖所有對儲存貯體的連線。如需有關如何修復此問題的資訊，請參閱我的儲存貯體政策有錯誤的 VPC 或 VPC 端點 ID。我該如何修復政策，讓我可以存取儲存貯體？(位於 支援 知識中心)。
> - 使用下列範例儲存貯體政策之前，請以適合您使用案例的適當值取代 VPC 端點 ID。否則，您將無法存取儲存貯體。
> - 如果您的政策僅允許從特定 VPC 端點存取 S3 on Outposts 儲存貯體，則它會停用對該儲存貯體的主控台存取權，因為主控台請求不是源自指定的 VPC 端點。

主題

範例：限制從 VPC 端點對特定儲存貯體的存取

您可以建立端點政策，以限制只存取特定 S3 on Outposts 儲存貯體。以下政策將 GetBucketPolicy 動作的存取權僅限制於 *example-outpost-bucket*。若要使用這個政策，請使用您的值來取代範例值。

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909151",
  "Statement": [
    { "Sid": "Access-to-specific-bucket-only",
      "Principal": {"AWS":"111122223333"},
      "Action": "s3-outposts:GetBucketPolicy",
      "Effect": "Allow",
      "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-
bucket"
    }
  ]
}
```

範例：在 S3 on Outposts 儲存貯體政策中拒絕從特定 VPC 端點存取

以下 S3 on Outposts 儲存貯體政策拒絕透過 *vpce-1a2b3c4d* VPC 端點在 *example-outpost-bucket* 儲存貯體上存取 GetBucketPolicy。

`aws:sourceVpce` 條件會指定端點，且不需要 VPC 端點資源的 Amazon Resource Name (ARN)，只需要端點 ID。若要使用這個政策，請使用您的值來取代範例值。

```
{
    "Version": "2012-10-17",
    "Id": "Policy1415115909152",
    "Statement": [
        {
            "Sid": "Deny-access-to-specific-VPCE",
            "Principal": {"AWS":"111122223333"},
            "Action": "s3-outposts:GetBucketPolicy",
            "Effect": "Deny",
            "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-
bucket",
            "Condition": {
```

```
                "StringEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}
            }
        }
    ]
}
```

AWS Signature 第 4 版 (SigV4) 身分驗證特定政策金鑰

下表顯示與 AWS Signature 第 4 版 (SigV4) 身分驗證相關的條件金鑰，您可以搭配 Amazon S3 on Outposts 使用。在儲存貯體政策中，您可以新增這些條件，以便在使用第 4 版簽署程序來驗證請求時強制執行特定行為。如需範例政策，請參閱 使用第 4 版簽署程序相關條件金鑰的儲存貯體政策範例。如需使用 Signature 第 4 版驗證請求的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的驗證請求 (AWS Signature 第 4 版)

適用金鑰	描述

`s3-outposts:authType`	S3 on Outposts 支援多種不同的身分驗證方法。若要限制傳入請求使用特定身分驗證方法，您可以使用此可選條件金鑰。例如，您可以使用此條件金鑰，僅允許 HTTP `Authorization` 標頭用在請求身分驗證中。
	
	有效值：
	
	`REST-HEADER`
	
	`REST-QUERY-STRING`

`s3-outposts:signatureAge`	簽章在已驗證請求中有效的時間長度 (以毫秒為單位)。 此條件僅適用於預先簽章 URL。 在 第 4 版簽署程序中，簽署金鑰的有效期限最長七天。因此，簽章的有效期限也是最長七天。如需詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的「簽署請求簡介」。您可以使用此條件來進一步限制簽章存留期。 範例值：`600000`

適用金鑰	描述
s3-outposts:x-amz-content-sha256	您可以使用此條件金鑰以不允許在儲存貯體中未簽署的內容。 當您使用第 4 版簽署程序時，針對使用 Authorization 標頭的請求，會在簽署計算中新增 x-amz-content-sha256 標頭，然後將其值設定為雜湊承載。 您可以在儲存貯體政策中使用此條件金鑰，拒絕任何尚未簽署承載的上傳項目。例如： • 拒絕使用了 Authorization 標頭來驗證請求但並未簽署承載的上傳項目。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的在單個區塊中傳輸承載。 • 拒絕使用預先簽章 URL 的上傳。預先簽章 URL 一律有 UNSIGNED_PAYLOAD 。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的身分驗證請求和身分驗證方法。 有效值：UNSIGNED-PAYLOAD

使用第 4 版簽署程序相關條件金鑰的儲存貯體政策範例

若要使用下列範例，請以您自己的資訊取代 *user input placeholders*。

Example : **s3-outposts:signatureAge**

下列儲存貯體政策會拒絕 example-outpost-bucket 中物件上的任何 S3 on Outposts 預先簽章 URL 請求 (如果簽章已超過 10 分鐘之久)。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Deny a presigned URL request if the signature is more than 10
 minutes old",
            "Effect": "Deny",
            "Principal": {"AWS":"444455556666"},
            "Action": "s3-outposts:*",
```

```
            "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
            "Condition": {
                "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
                "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
            }
        }
    ]
}
```

Example : **s3-outposts:authType**

下列儲存貯體政策僅允許使用 Authorization 標頭以供請求驗證的任何請求。任何預先簽章 URL 請求都會遭到拒絕，因為預先簽章 URL 會使用查詢參數來提供請求和驗證資訊。如需詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的「身分驗證方法」。

```
{
   "Version": "2012-10-17",
   "Statement": [
         {
               "Sid": "Allow only requests that use the Authorization header for
 request authentication. Deny presigned URL requests.",
               "Effect": "Deny",
               "Principal": {"AWS":"111122223333"},
               "Action": "s3-outposts:*",
               "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
               "Condition": {
                     "StringNotEquals": {
                           "s3-outposts:authType": "REST-HEADER"
                     }
               }
         }
   ]
}
```

Example : **s3-outposts:x-amz-content-sha256**

下列儲存貯體政策會拒絕任何具有未簽署承載的上傳項目，例如使用預先簽章 URL 的上傳。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的身分驗證請求和身分驗證方法。

```
{
   "Version": "2012-10-17",
   "Statement": [
         {
               "Sid": "Deny uploads with unsigned payloads.",
               "Effect": "Deny",
               "Principal": {"AWS":"111122223333"},
               "Action": "s3-outposts:*",
               "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
               "Condition": {
                     "StringEquals": {
                           "s3-outposts:x-amz-content-sha256": "UNSIGNED-PAYLOAD"
                     }
               }
         }
   ]
}
```

AWS Amazon S3 on Outposts 的 受管政策

AWS 受管政策是由 AWS AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住， AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新受 AWS 管政策中定義的許可，則更新會影響政策連接的所有主體身分（使用者、群組和角色）。 AWS 服務 當新的 啟動或新的 API 操作可用於現有服務時， AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 受管政策：AWSS3OnOutpostsServiceRolePolicy

作為服務連結角色 `AWSServiceRoleForS3OnOutposts` 的一部分，協助您管理網路資源。

若要檢視此政策的許可，請參閱 AWSS3OnOutpostsServiceRolePolicy。

AWS 受管政策的 S3 on Outposts 更新

檢視自此服務開始追蹤這些變更以來S3 on Outposts AWS 受管政策更新的詳細資訊。

變更	描述	日期
S3 on Outposts 新增了 `AWSS3OnOutpostsServiceRolePolicy`	S3 on Outposts 新增了 `AWSS3OnOutpostsServiceRolePolicy` 作為服務連結角色 `AWSServiceRoleForS3OnOutposts` 的一部分，可協助您管理網路資源。	2023 年 10 月 3 日
S3 on Outposts 開始追蹤變更	S3 on Outposts 開始追蹤其 AWS 受管政策的變更。	2023 年 10 月 3 日

針對 Amazon S3 on Outposts 使用服務連結角色

Amazon S3 on Outposts 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 S3 on Outposts 的一種特殊 IAM 角色類型。服務連結角色是由 S3 on Outposts 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓設定 S3 on Outposts 更為容易，因為您不必手動新增必要的許可。S3 on Outposts 會定義其服務連結角色的許可，除非另有定義，否則只有 S3 on Outposts 可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除服務連結角色的相關資源，才能將其刪除。如此可保護您 S3 on Outposts 資源，避免您不小心移除存取資源的許可。

如需支援服務連結角色的其他 服務的資訊，請參閱 AWS 服務與 IAM 搭配使用，並在服務連結角色欄中尋找具有是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

S3 on Outposts 的服務連結角色許可

S3 on Outposts 使用名為 AWSServiceRoleForS3OnOutposts 的服務連結角色來協助您管理網路資源。

`AWSServiceRoleForS3OnOutposts` 服務連結角色信任下列服務以擔任角色：

- s3-outposts.amazonaws.com

名為 AWSS3OnOutpostsServiceRolePolicy 的角色許可政策允許 S3 on Outposts 對指定的資源完成下列動作：

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeCoipPools",
                "ec2:GetCoipPoolUsage",
                "ec2:DescribeAddresses",
                "ec2:DescribeLocalGatewayRouteTableVpcAssociations"
            ],
            "Resource": "*",
            "Sid": "DescribeVpcResources"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ],
            "Sid": "CreateNetworkInterface"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ],
            "Condition": {
                "StringEquals": {
```

```
                    "aws:RequestTag/CreatedBy": "S3 On Outposts"
                }
            },
            "Sid": "CreateTagsForCreateNetworkInterface"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AllocateAddress"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:ipv4pool-ec2/*"
            ],
            "Sid": "AllocateIpAddress"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AllocateAddress"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:elastic-ip/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CreatedBy": "S3 On Outposts"
                }
            },
            "Sid": "CreateTagsForAllocateIpAddress"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DisassociateAddress",
                "ec2:ReleaseAddress",
                "ec2:AssociateAddress"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:ResourceTag/CreatedBy": "S3 On Outposts"
                }
            },
            "Sid": "ReleaseVpcResources"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "CreateNetworkInterface",
                        "AllocateAddress"
                    ],
                    "aws:RequestTag/CreatedBy": [
                        "S3 On Outposts"
                    ]
                }
            },
            "Sid": "CreateTags"
        }
    ]
}
```

您必須設定許可，IAM 實體 (如角色) 才能建立、編輯或刪除服務連結角色。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。

為 S3 on Outposts 建立服務連結角色

您不需要手動建立一個服務連結角色。當您在 AWS Management Console、 AWS CLI或 AWS API 中建立 S3 on Outposts 端點時，S3 on Outposts 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 S3 on Outposts 端點時，S3 on Outposts 會再次為您建立服務連結角色。

您也可以使用 IAM 主控台，參考 S3 on Outposts 使用案例來建立服務連結角色。在 AWS CLI 或 AWS API 中，使用服務名稱建立`s3-outposts.amazonaws.com`服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的「建立服務連結角色」。如果您刪除此服務連結角色，您可以使用此相同的程序以再次建立該角色。

編輯 S3 on Outposts 的服務連結角色

S3 on Outposts 不允許您編輯 AWSServiceRoleForS3OnOutposts 服務連結角色。包括角色的名稱也不可編輯，因為可能有各種不同的實體參考角色。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱「IAM 使用者指南」的編輯服務連結角色。

刪除 S3 on Outposts 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。然而，在手動刪除服務連結角色之前，您必須先清除資源。

> **Note**
>
> 若 S3 on Outposts 服務在您試圖刪除資源時正在使用該角色，刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

刪除 AWSServiceRoleForS3OnOutposts 角色使用的 S3 on Outposts 資源

1. AWS 帳戶 在所有 中刪除 S3 on Outposts 端點 AWS 區域。
2. 使用 IAM 刪除服務連結角色。

   使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除AWSServiceRoleForS3OnOutposts服務連結角色。如需詳細資訊，請參閱「IAM 使用者指南」中的刪除服務連結角色。

S3 on Outposts 服務連結角色的支援區域

S3 on Outposts 支援在所有提供服務 AWS 區域 的 中使用服務連結角色。如需詳細資訊，請參閱 S3 on Outposts 區域和端點。

管理 S3 on Outposts 儲存貯體

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (0UTP0STS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

如需管理和共享 Amazon S3 on Outposts 儲存貯體容量的詳細資訊，請參閱下列主題。

主題

- 針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制
- 建立和管理 Amazon S3 on Outposts 儲存貯體的生命週期組態
- 複寫 S3 on Outposts 的物件
- 使用 共用 S3 on Outposts AWS RAM
- 使用 S3 on Outposts AWS 服務 的其他

針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制

啟用時，S3 版本控制會在相同的儲存貯體中儲存物件的多個不同複本。您可以使用 S3 版本控制，保留、擷取和還原在 Outposts 儲存貯體中所存放每個物件的各個版本。S3 版本控制可協助您從意外的使用者動作和應用程式失敗中復原。

Amazon S3 on Outposts 儲存貯體具有三種版本控制狀態：

- Unversioned (未版本控制) - 如果您從未在儲存貯體上啟用或暫停 S3 版本控制，則表示未版本控制，並且不會傳回任何 S3 版本控制狀態。如需 S3 版本控制的詳細資訊，請參閱「針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制」。
- Enabled (已啟用) - 針對儲存貯體中的物件啟用 S3 版本控制。所有新增至儲存貯體的物件都會收到唯一的版本 ID。啟用版本控制時已存在於儲存貯體中的物件的版本 ID 為 `null`。如果您使用其他操作修改這些 (或任何其他) 物件，例如 PutObject，新物件會取得唯一的版本 ID。

- Suspended (已暫停) - 針對儲存貯體中的物件暫停 S3 版本控制。所有在版本控制暫停之後新增至儲存貯體的物件都會收到版本 ID `null`。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的將物件新增至暫停版本控制的儲存貯體。

在您針對 S3 on Outposts 儲存貯體啟用 S3 版本控制之後，此儲存貯體永遠無法回復為未使用版本控制狀態。不過，您可以暫停版本控制。如需 S3 版本控制的詳細資訊，請參閱「針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制」。

對於儲存貯體中的每個物件，您都有一個目前版本，以及零個以上的非目前版本。若要降低儲存成本，您可以將儲存貯體 S3 生命週期規則設定為在指定的時段之後使非目前版本過期。如需詳細資訊，請參閱建立和管理 Amazon S3 on Outposts 儲存貯體的生命週期組態。

下列範例說明如何使用 AWS Management Console 和 AWS Command Line Interface () 來啟用或停用現有 S3 on Outposts 儲存貯體的版本控制AWS CLI。若要建立已啟用版本控制的 S3 儲存貯體，請參閱 建立 S3 on Outposts 儲存貯體。

> **Note**
>
> 建立儲存貯體 AWS 帳戶 的 擁有該儲存貯體，是唯一可以對其遞交動作的 。儲存貯體具有組態屬性，例如 Outpost、標籤、預設加密和存取點設定。存取點設定包含用於存取儲存貯體中物件的虛擬私有雲端 (VPC) 和存取點政策，以及其他中繼資料。如需詳細資訊，請參閱「S3 on Outposts 規格」。

使用 S3 主控台

編輯儲存貯體的 S3 版本控制設定

1. 登入 AWS Management Console 並開啟位於 https：//Amazon S3 主控台。 https://console.aws.amazon.com/s3/
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要針對其啟用 S3 版本控制的 Outposts 儲存貯體。
4. 選擇屬性索引標籤。
5. 在 Bucket Versioning (儲存貯體版本控制) 底下，選擇 Edit (編輯)。
6. 選擇下列其中一個選項來編輯儲存貯體的 S3 版本控制設定：

   - 若要暫停 S3 版本控制並停止建立新的物件版本，請選擇 Suspend (暫停)。

- 若要啟用 S3 版本控制並儲存每個物件的多個不同複本，請選擇 Enable (啟用)。

7. 選擇 Save changes (儲存變更)。

使用 AWS CLI

若要使用 啟用或停用儲存貯體的 S3 版本控制 AWS CLI，請使用 `put-bucket-versioning`命令，如下列範例所示。若要使用這些範例，請以您自己的資訊取代每個 *`user input placeholder`*。

如需詳細資訊，請參閱《AWS CLI 參考》中的 put-bucket-versioning。

Example ：啟用 S3 版本控制

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --versioning-configuration Status=Enabled
```

Example ：暫停 S3 版本控制

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --versioning-configuration Status=Suspended
```

建立和管理 Amazon S3 on Outposts 儲存貯體的生命週期組態

您可以使用 S3 生命週期，最佳化 Amazon S3 on Outposts 的儲存容量。您可以建立生命週期規則，在物件老化或取代為較新的版本時使這些物件過期。您可以建立、啟用、停用或刪除生命週期規則。

如需 S3 生命週期的詳細資訊，請參閱 建立和管理 Amazon S3 on Outposts 儲存貯體的生命週期組態。

> **Note**
>
> 建立儲存貯體 AWS 帳戶 的 擁有它，而且是唯一可以建立、啟用、停用或刪除生命週期規則的 。

若要為您的 S3 on Outposts 儲存貯體建立和管理生命週期組態，請參閱下列主題。

主題

- 使用 建立和管理生命週期規則 AWS Management Console
- 使用適用於 Java 的 AWS CLI 和 SDK 建立和管理生命週期組態

使用 建立和管理生命週期規則 AWS Management Console

您可以使用 S3 生命週期，最佳化 Amazon S3 on Outposts 的儲存容量。您可以建立生命週期規則，在物件老化或取代為較新的版本時使這些物件過期。您可以建立、啟用、停用或刪除生命週期規則。

如需 S3 生命週期的詳細資訊，請參閱 建立和管理 Amazon S3 on Outposts 儲存貯體的生命週期組態。

> **Note**
>
> 建立儲存貯體 AWS 帳戶 的 擁有它，而且是唯一可以建立、啟用、停用或刪除生命週期規則的 。

若要使用 建立和管理 S3 on Outposts 的生命週期規則 AWS Management Console，請參閱下列主題。

主題

- 建立生命週期規則
- 啟用生命週期規則
- 編輯生命週期規則
- 刪除生命週期規則

建立生命週期規則

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台開啟 https://console.aws.amazon.com/s3/ S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要為其建立生命週期規則的 Outposts 儲存貯體。
4. 選擇 Management (管理) 索引標籤，然後選擇 Create lifecycle rule (建立生命週期規則)。
5. 輸入 Lifecycle rule name (生命週期規則名稱) 的值。
6. 在 Rule scope (規則範圍) 下，選擇下列其中一個選項：

- 若要將範圍限制為特定篩選條件，請選擇 Limit the scope of this rule using one or more filters (使用一或多個篩選條件限制此規則的範圍)。然後，新增字首篩選條件、索引標籤或物件大小。
- 若要將規則套用至儲存貯體中的所有物件，請選擇 Apply to all objects in the bucket (套用至儲存貯體中的所有物件)。

7. 在 Lifecycle rule actions (生命週期規則動作) 下，選擇下列其中一個選項：

   - Expire current versions of objects (讓目前版本的物件過期) - 對於已啟用版本控制的儲存貯體，S3 on Outposts 會新增刪除標記，並將物件保留為非目前版本。對於未使用 S3 版本控制的儲存貯體，S3 on Outposts 會永久刪除這些物件。
   - Permanently delete noncurrent versions of objects (永久刪除非目前版本的物件) - S3 on Outposts 會永久刪除非目前版本的物件。
   - Delete expired object delete markers or incomplete multipart uploads (刪除過期物件刪除標記或未完成的分段上傳) - S3 on Outposts 會永久刪除過期物件刪除標記或未完成的分段上傳。

     如果您使用物件標籤來限制生命週期規則的範圍，則無法選擇 Delete expired object delete markers (刪除過期物件刪除標記)。如果您選擇 Expire current object versions (讓目前版本的物件過期)，也無法選擇 Delete expired object delete markers (刪除過期物件刪除標記)。

   > **Note**
   >
   > 大小型篩選條件無法與刪除標記和未完成的分段上傳搭配使用。

8. 如果您選擇 Expire current versions of objects (讓目前版本的物件過期) 或 Permanently delete noncurrent versions of objects (永久刪除非目前版本的物件)，請根據特定日期或物件的存留期來設定規則觸發條件。
9. 如果您選擇了 Delete expired object delete markers (刪除過期物件刪除標記)，為了確認您想要刪除過期物件刪除標記，請選取 Delete expired object delete markers (刪除過期物件刪除標記)。
10. 在 Timeline Summary (時間軸摘要) 下，檢閱您的生命週期規則，然後選擇 Create rule (建立規則)。

啟用生命週期規則

若要啟用或停用儲存貯體生命週期規則

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。

2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要啟用或停用其生命週期規則的 Outposts 儲存貯體。
4. 選擇 Management (管理) 索引標籤，然後在 Lifecycle rule (生命週期規則) 下，選擇您要啟用或停用的規則。
5. 對於 Action (動作)，選擇 Enable or disable rule (啟用或停用規則)。

編輯生命週期規則

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要為其編輯生命週期規則的 Outposts 儲存貯體。
4. 選擇 Management (管理) 索引標籤，然後選擇您要編輯的 Lifecycle rule (生命週期規則)。
5. (選用) 更新 Lifecycle rule name (生命週期規則名稱) 的值。
6. 在 Rule scope (規則範圍) 下，視需要編輯範圍：
   - 若要將範圍限制為特定篩選條件，請選擇 Limit the scope of this rule using one or more filters (使用一或多個篩選條件限制此規則的範圍)。然後，新增字首篩選條件、索引標籤或物件大小。
   - 若要將規則套用至儲存貯體中的所有物件，請選擇 Apply to all objects in the bucket (套用至儲存貯體中的所有物件)。
7. 在 Lifecycle rule actions (生命週期規則動作) 下，選擇下列其中一個選項：
   - Expire current versions of objects (讓目前版本的物件過期) - 對於已啟用版本控制的儲存貯體，S3 on Outposts 會新增刪除標記，並將物件保留為非目前版本。對於未使用 S3 版本控制的儲存貯體，S3 on Outposts 會永久刪除這些物件。
   - Permanently delete noncurrent versions of objects (永久刪除非目前版本的物件) - S3 on Outposts 會永久刪除非目前版本的物件。
   - Delete expired object delete markers or incomplete multipart uploads (刪除過期物件刪除標記或未完成的分段上傳) - S3 on Outposts 會永久刪除過期物件刪除標記或未完成的分段上傳。

     如果您使用物件標籤來限制生命週期規則的範圍，則無法選擇 Delete expired object delete markers (刪除過期物件刪除標記)。如果您選擇 Expire current object versions (讓目前版本的物件過期)，也無法選擇 Delete expired object delete markers (刪除過期物件刪除標記)。

> **Note**
>
> 大小型篩選條件無法與刪除標記和未完成的分段上傳搭配使用。

8. 如果您選擇 Expire current versions of objects (讓目前版本的物件過期)或 Permanently delete noncurrent versions of objects (永久刪除非目前版本的物件)，請根據特定日期或物件存留期來設定規則觸發條件。
9. 如果您選擇了 Delete expired object delete markers (刪除過期物件刪除標記)，為了確認您想要刪除過期物件刪除標記，請選取 Delete expired object delete markers (刪除過期物件刪除標記)。
10. 選擇 Save (儲存)。

刪除生命週期規則

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要為其刪除生命週期規則的 Outposts 儲存貯體。
4. 選擇 Management (管理) 索引標籤，然後在 Lifecycle rule (生命週期規則)，選擇您要刪除的規則。
5. 選擇 Delete (刪除)。

使用適用於 Java 的 AWS CLI 和 SDK 建立和管理生命週期組態

您可以使用 S3 生命週期，最佳化 Amazon S3 on Outposts 的儲存容量。您可以建立生命週期規則，在物件老化或取代為較新的版本時使這些物件過期。您可以建立、啟用、停用或刪除生命週期規則。

如需 S3 生命週期的詳細資訊，請參閱 建立和管理 Amazon S3 on Outposts 儲存貯體的生命週期組態。

> **Note**
>
> 建立儲存貯體 AWS 帳戶 的 擁有它，而且是唯一可以建立、啟用、停用或刪除生命週期規則的 。

若要使用 AWS Command Line Interface (AWS CLI) 和 建立和管理 S3 on Outposts 儲存貯體的生命週期組態 適用於 Java 的 AWS SDK，請參閱下列範例。

主題

放置生命週期組態

AWS CLI

下列 AWS CLI 範例會在 Outposts 儲存貯體上放置生命週期組態政策。此政策指定，所有包含標記字首 (*myprefix*) 和標籤的物件會在 10 天後過期。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。

1. 將生命週期組態原則儲存至 JSON 檔案。在此範例中，檔案命名為 `lifecycle1.json`。

```
{
    "Rules": [
        {
            "ID": "id-1",
            "Filter": {
                "And": {
                    "Prefix": "myprefix",
                    "Tags": [
                        {
                            "Value": "mytagvalue1",
                            "Key": "mytagkey1"
                        },
                        {
                            "Value": "mytagvalue2",
                            "Key": "mytagkey2"
                        }
                    ],
                    "ObjectSizeGreaterThan": 1000,
                    "ObjectSizeLessThan": 5000
                }
            },
            "Status": "Enabled",
            "Expiration": {
                "Days": 10
```

```
            }
        }
    ]
}
```

2. 提交 JSON 檔案以做為 put-bucket-lifecycle-configuration CLI 命令的一部分。若要執行此命令，請以您自己的資訊取代每個 *user input placeholder*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 put-bucket-lifecycle-configuration。

```
aws s3control put-bucket-lifecycle-configuration --account-id 123456789012 --
bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outposts-bucket --lifecycle-configuration file://lifecycle1.json
```

SDK for Java

下列適用於 Java 的開發套件範例將生命週期組態放置在 Outpost 儲存貯體上。此生命週期組態指定，所有包含標記字首 (*myprefix*) 和標籤的物件會在 10 天後過期。若要使用此範例，請以您自己的資訊取代每個 *user input placeholder*。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 PutBucketLifecycleConfiguration。

```
import com.amazonaws.services.s3control.model.*;

public void putBucketLifecycleConfiguration(String bucketArn) {

    S3Tag tag1 = new S3Tag().withKey("mytagkey1").withValue("mytagkey1");
    S3Tag tag2 = new S3Tag().withKey("mytagkey2").withValue("mytagkey2");

    LifecycleRuleFilter lifecycleRuleFilter = new LifecycleRuleFilter()
            .withAnd(new LifecycleRuleAndOperator()
                    .withPrefix("myprefix")
                    .withTags(tag1, tag2))
                    .withObjectSizeGreaterThan(1000)
                    .withObjectSizeLessThan(5000);

    LifecycleExpiration lifecycleExpiration = new LifecycleExpiration()
            .withExpiredObjectDeleteMarker(false)
            .withDays(10);

    LifecycleRule lifecycleRule = new LifecycleRule()
            .withStatus("Enabled")
            .withFilter(lifecycleRuleFilter)
```

```
            .withExpiration(lifecycleExpiration)
            .withID("id-1");

    LifecycleConfiguration lifecycleConfiguration = new LifecycleConfiguration()
            .withRules(lifecycleRule);

    PutBucketLifecycleConfigurationRequest reqPutBucketLifecycle = new
 PutBucketLifecycleConfigurationRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn)
            .withLifecycleConfiguration(lifecycleConfiguration);

    PutBucketLifecycleConfigurationResult respPutBucketLifecycle =
 s3ControlClient.putBucketLifecycleConfiguration(reqPutBucketLifecycle);
    System.out.printf("PutBucketLifecycleConfiguration Response: %s%n",
 respPutBucketLifecycle.toString());

}
```

取得 S3 on Outposts 儲存貯體的生命週期組態

AWS CLI

下列 AWS CLI 範例取得 Outposts 儲存貯體的生命週期組態。若要執行此命令，請以您自己的資訊取代每個 *user input placeholder*。如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 get-bucket-lifecycle-configuration。

```
aws s3control get-bucket-lifecycle-configuration --account-id 123456789012 --bucket
 arn:aws:s3-outposts:<your-region>:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outposts-bucket
```

SDK for Java

下列適用於 Java 的開發套件範例取得 Outpost 儲存貯體的生命週期組態。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 GetBucketLifecycleConfiguration。

```
import com.amazonaws.services.s3control.model.*;

public void getBucketLifecycleConfiguration(String bucketArn) {

    GetBucketLifecycleConfigurationRequest reqGetBucketLifecycle = new
 GetBucketLifecycleConfigurationRequest()
```

```
            .withAccountId(AccountId)
            .withBucket(bucketArn);

    GetBucketLifecycleConfigurationResult respGetBucketLifecycle =
 s3ControlClient.getBucketLifecycleConfiguration(reqGetBucketLifecycle);
    System.out.printf("GetBucketLifecycleConfiguration Response: %s%n",
 respGetBucketLifecycle.toString());

}
```

複寫 S3 on Outposts 的物件

開啟 S3 複寫時 AWS Outposts，您可以設定 Amazon S3 on Outposts 以自動複寫不同 Outposts 或相同 Outpost 上儲存貯體之間的 S3 物件。您可以使用 S3 Replication on Outposts，在相同或不同的 Outposts 中或跨不同的帳戶維護資料的多個複本，以協助符合資料駐留需求。S3 Replication on Outposts 有助於支援合規儲存需求，以及跨帳戶資料共用。如果需要確保複本與來源資料相同，您可以使用 S3 Replication on Outposts 來建立保留所有中繼資料的物件複本，例如原始物件建立時間、標籤和版本 ID。

S3 Replication on Outposts 也提供詳細的指標和通知，以監控儲存貯體之間的物件複寫狀態。您可以使用 Amazon CloudWatch 監控複寫進度，方法為追蹤位元組等待複寫、操作等待複寫，以及來源與目標儲存貯體之間的複寫延遲。若要快速診斷並更正組態問題，您也可以將 Amazon EventBridge 設定為接收有關複寫物件失敗的通知。如需進一步了解，請參閱 管理複寫。

主題

複寫組態

S3 on Outposts 會以 XML 的形式存放複寫組態。在複寫組態 XML 檔案中，您可以指定 AWS Identity and Access Management (IAM) 角色和一或多個規則。

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
</ReplicationConfiguration>
```

若沒有您的許可，S3 on Outposts 無法複寫物件。您可以使用複寫組態中指定的 IAM 角色來授予 S3 on Outposts 許可。S3 on Outposts 會擔任該 IAM 角色來代您複寫物件。您必須將所需的許可授予 IAM 角色，然後才能開始複寫。如需這些 S3 on Outposts 許可的詳細資訊，請參閱 建立 IAM 角色。

針對下列情況，您可以在複寫組態中新增一個規則：

- 您想要複寫所有物件。
- 您想要複寫物件子集。您在規則中新增篩選條件，以識別物件子集。您可以在篩選條件中指定物件金鑰前綴、標籤，或這兩項的組合，以識別要套用規則的物件子集。

若您想複寫不同的物件子集，可以在複寫組態中新增多項規則。在每個規則中，您可以指定篩選條件以選取不同的物件子集。例如，您可以選擇複寫含有 `tax/` 或 `document/` 索引鍵字首的物件。要做到這一點，您需要新增兩個規則，一個指定 `tax/` 索引鍵字首篩選條件，另一個指定 `document/` 索引鍵字首。

如需 S3 on Outposts 複寫組態和複寫規則的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的ReplicationConfiguration。

S3 Replication on Outposts 的需求

複寫需求如下：

- 目的地 Outpost CIDR 範圍必須與來源 Outpost 子網路表格相關聯。如需詳細資訊，請參閱建立複寫規則的先決條件。

- 來源與目的地儲存貯體都必須啟用 S3 版本控制。如需版本控制的詳細資訊，請參閱「針對您的 S3 on Outposts 儲存貯體管理 S3 版本控制」。
- Amazon S3 on Outposts 必須具備許可，才能代您將物件從來源儲存貯體複寫至目的地儲存貯體。這意味著您必須建立一個服務角色，將 GET 和 PUT 許可委派給 S3 on Outposts。
  1. 在建立服務角色之前，您必須對來源儲存貯體具有 GET 許可，以及對目的地儲存貯體具有 PUT 許可。
  2. 若要建立服務角色以將許可委派給 S3 on Outposts，您必須先設定許可，以允許 IAM 實體 (使用者或角色) 執行 `iam:CreateRole` 和 `iam:PassRole` 動作。然後，允取 IAM 實體建立服務角色。若要讓 S3 on Outposts 代您擔任服務角色，並將 GET 和 PUT 許可委派給 S3 on Outposts，您必須將所需的信任和許可政策指派給該角色。如需這些 S3 on Outposts 許可的詳細資訊，請參閱 建立 IAM 角色。如需建立服務角色的詳細資訊，請參閱建立服務角色。

複寫內容為何？

依預設，S3 on Outposts 會複寫下列項目：

- 在您新增複寫組態之後建立的物件。
- 從來源物件到複本的物件中繼資料。如需如何將中繼資料從複本複寫至來源物件的相關資訊，請參閱啟用 Outposts 上的 Amazon S3 複本修改同步時的複寫狀態。
- 物件標籤 (如果有)。

刪除操作對複寫的影響

如果您從來源儲存貯體中刪除物件，預設會執行下列動作：

- 如果您發出 DELETE 請求但未指定物件版本 ID，則 S3 on Outposts 會新增刪除標記。S3 on Outposts 會如下處理刪除標記：
  - S3 on Outposts 預設不會複寫刪除標記。
  - 但是，您可以將刪除標記複寫新增至非標籤型規則。如需如何在複寫組態中啟用刪除標記複寫的詳細資訊，請參閱使用 S3 主控台。
- 如果您在 DELETE 請求中指定要刪除的物件版本 ID，S3 on Outposts 會永久刪除來源儲存貯體中的該物件版本。不過，不會在目的地儲存貯體中複寫刪除。換句話說，Amazon S3 不會從目的地儲存貯體中刪除相同的物件版本。此行為可防止資料遭到惡意刪除。

未複寫內容為何？

依預設，S3 on Outposts 不會複寫下列項目：

- 來源儲存貯體中由其他複寫規則所建立的物件複本。例如，若您設定複寫，其中儲存貯體 A 是來源，而儲存貯體 B 是目的地。現在，假設您新增另一個複寫組態，其中儲存貯體 B 是來源，而儲存貯體 C 是目的地。在此情況下，如果儲存貯體 B 中的物件是儲存貯體 A 中的物件複本，則不會複寫至儲存貯體 C。
- 來源儲存貯體中已複寫至不同目的地的物件。例如，如果您變更現有複寫組態中的目的地儲存貯體，則 S3 on Outposts 不會再次複寫這些物件。
- 在使用客戶所提供加密金鑰 (SSE-C) 進行伺服器端加密情況下建立的物件。
- 儲存貯體層級子資源的更新。

  例如，如果您變更來源儲存貯體上的生命週期組態，或將通知組態新增至來源儲存貯體，這些變更並不會套用至目的地儲存貯體。此功能可讓來源儲存貯體與目的地儲存貯體各有不同的組態。
- 生命週期組態執行的動作。

  例如，如果您只在來源儲存貯體上啟用生命週期組態，並設定到期動作，則 S3 on Outposts 會為過期物件建立刪除標記，但不會將這些標記複寫至目的地儲存貯體。如果您想要將相同的生命週期組態套用至來源與目的地儲存貯體，請在這兩個儲存貯體上啟用相同的生命週期組態。如需生命週期組態的詳細資訊，請參閱「建立和管理 Amazon S3 on Outposts 儲存貯體的生命週期組態」。

S3 Replication on Outposts 不支援哪些項目？

S3 on Outposts 目前不支援下列 S3 複寫功能。

- S3 複寫時間控制 (S3 RTC)。不支援 S3 RTC，因為 S3 Replication on Outposts 中的物件流量會透過內部部署網路 (本機閘道) 傳輸。如需本機閘道的相關資訊，請參閱《AWS Outposts 使用者指南》中的使用本機閘道。
- 適用於批次操作的 S3 複寫。

設定複寫

> **Note**
>
> 您設定複寫之前就已存在於儲存貯體中的物件不會自動複寫。換句話說，Amazon S3 on Outposts 不會追溯複寫物件。若要複寫您在複寫組態之前建立的物件，您可使用

`CopyObject` API 操作以將其複製到相同的儲存貯體。複製物件之後，這些物件會在儲存貯體中顯示為「新」物件，且複寫組態將套用至這些物件。如需複製物件的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的 使用 在 Amazon S3 on Outposts 儲存貯體中複製物件 適用於 Java 的 AWS SDK 和 CopyObject。

若要啟用 S3 Replication on Outposts，請將複寫規則新增至來源 Outposts 儲存貯體。複寫規則會通知 S3 on Outposts 依指定方式複寫物件。在複寫規則中，您必須提供以下項目：

- 來源 Outposts 儲存貯體存取點 — 您想要 S3 on Outposts 從其中複寫物件的存取點 Amazon Resource Name (ARN) 或存取點別名。如需使用存取點別名的詳細資訊，請參閱針對您的 S3 on Outposts 存取點使用儲存貯體樣式別名。
- 您要複寫的物件 – 您可以複寫來源 Outposts 儲存貯體中的所有物件或物件子集。您可以在組態中提供金鑰名稱前綴、一或多個物件標籤或兩者，來識別子集。

  例如，如果您將複寫規則設定為僅複寫具有金鑰名稱前綴為 `Tax/` 的物件，則 S3 on Outposts 會複寫具有 `Tax/doc1` 或 `Tax/doc2` 等金鑰的物件。但不會複寫具有 `Legal/doc3` 金鑰的物件。如果您指定字首以及一或多個標籤，則 S3 on Outposts 僅會複寫具備特定索引鍵字首和標籤的物件。
- 目的地 Outposts 儲存貯體 — 您想要 S3 on Outposts 複寫物件的儲存貯體 ARN 或存取點別名。

您可以使用 REST API、 AWS SDKs、 AWS Command Line Interface (AWS CLI) 或 Amazon S3 主控台來設定複寫規則。

S3 on Outposts 也提供 API 操作來支援設定複寫規則。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的下列主題：

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

主題

- 建立複寫規則的先決條件
- 建立 Outposts 上的複寫規則

建立複寫規則的先決條件

主題

連線您的來源和目的地 Outpost 子網路

若要讓您的複寫流量透過本機閘道從來源 Outpost 傳送到目的地 Outpost，您必須新增新路由來設定網路。您必須將存取點的無類別域間路由 (CIDR) 網路範圍連接在一起。對於每對存取點，您只需要設定一次此連線。

根據與存取點相關聯 Outposts 端點的存取類型，設定連線的某些步驟會有所不同。端點的存取類型為私有（直接虛擬私有雲端 【VPC】 路由 AWS Outposts) 或客戶擁有的 IP（內部部署網路中的客戶擁有 IP 地址集區 【CoIP 集區】)。

步驟 1：尋找來源 Outposts 端點的 CIDR 範圍

尋找與來源存取點相關聯來源端點的 CIDR 範圍

1. 登入 AWS Management Console 並開啟位於 https：//Amazon S3 主控台。 https://console.aws.amazon.com/s3/
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 在 Outposts 儲存貯體清單中，選擇您要進行複寫的來源儲存貯體。
4. 選擇 Outposts 存取點索引標籤，然後針對複寫規則選擇來源儲存貯體的 Outposts 存取點。
5. 選擇 Outposts 端點。
6. 複製要在步驟 5 中使用的子網路 ID。
7. 您用來尋找來源 Outposts 端點 CIDR 範圍的方法取決於端點的存取類型。

   在 Outposts 端點概觀區段中，請參閱存取類型。

   - 如果存取類型為私有，請複製要在步驟 6 中使用的無類別域間路由 (CIDR) 值。
   - 如果存取類型為客戶擁有的 IP，請執行下列動作：
     1. 複製客戶擁有的 IPv4 集區值，以便稍後使用做為地址集區的 ID。
     2. 開啟 AWS Outposts 主控台，網址為 https://console.aws.amazon.com/outposts/：//。
     3. 在導覽窗格中，選擇本機閘道路由表。

4. 選擇來源 Outpost 的本機閘道路由表 ID 值。
5. 在詳細資訊窗格中，選擇 CoIP 集區索引標籤。將先前所複製 CoIP 集區 ID 的值貼到搜尋方塊中。
6. 對於相符的 CoIP 集區，複製來源 Outposts 端點的對應 CIDR 值，以便在步驟 6 中使用。

步驟 2：尋找目的地 Outposts 端點的子網路 ID 和 CIDR 範圍

若要尋找與目的地存取點相關聯目的地端點的子網路 ID 和 CIDR 範圍，請遵循步驟 1 中的相同子步驟，並在套用這些子步驟時，將來源 Outposts 端點變更為目的地 Outposts 端點。複製目的地 Outposts 端點的子網路 ID 值，以便在步驟 6 中使用。複製目的地 Outposts 端點的 CIDR 值，以便在步驟 5 中使用。

步驟 3：尋找來源 Outpost 的本機閘道 ID

尋找來源 Outpost 的本機閘道 ID

1. 開啟 AWS Outposts 主控台，網址為 https://console.aws.amazon.com/outposts/：//。
2. 在左側導覽窗格中，選擇本機閘道。
3. 在本機閘道頁面上，尋找您要用於複寫來源 Outpost 的 Outpost ID。
4. 複製來源 Outpost 的本機閘道 ID 值，以便在步驟 5 中使用。

如需本機閘道的詳細資訊，請參閱《AWS Outposts 使用者指南》中的本機閘道。

步驟 4：尋找目的地 Outpost 的本機閘道 ID

若要尋找目的地 Outpost 的本機閘道 ID，請遵循步驟 3 中的相同子步驟，但尋找目的地 Outpost 的 Outpost ID 步驟除外。複製目的地 Outpost 的本機閘道 ID 值，以便在步驟 6 中使用。

步驟 5：設定從來源 Outpost 子網路到目的地 Outpost 子網路的連線

從來源 Outpost 子網路連線至目的地 Outpost 子網路

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/vpc/ 開啟 VPC 主控台。
2. 在左側導覽窗格中，選擇 Subnets (子網路)。
3. 在搜尋方塊中，輸入您在步驟 1 中所找到來源 Outposts 端點的子網路 ID。選擇具有相符子網路 ID 的子網路。
4. 對於相符的子網路項目，請選擇此子網路的路由表值。

5. 在具有所選路由表的頁面上，選擇動作，然後選擇編輯路由。
6. 在路由標籤中，選擇編輯路由。
7. 在目的地下，輸入您在步驟 2 中所找到目的地 Outposts 端點的 CIDR 範圍。
8. 在目標下，選擇 Outpost 本機閘道，然後輸入您在步驟 3 中所找到來源 Outpost 的本機閘道 ID。
9. 選擇 Save changes (儲存變更)。
10. 確定路由的狀態為作用中。

步驟 6：設定從目的地 Outpost 子網路到來源 Outpost 子網路的連線

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/vpc/ 開啟 VPC 主控台。
2. 在左側導覽窗格中，選擇 Subnets (子網路)。
3. 在搜尋方塊中，輸入您在步驟 2 中所找到目的地 Outposts 端點的子網路 ID。選擇具有相符子網路 ID 的子網路。
4. 對於相符的子網路項目，請選擇此子網路的路由表值。
5. 在具有所選路由表的頁面上，選擇動作，然後選擇編輯路由。
6. 在路由標籤中，選擇編輯路由。
7. 在目的地下，輸入您在步驟 1 中所找到來源 Outposts 端點的 CIDR 範圍。
8. 在目標下，選擇 Outpost 本機閘道，然後輸入您在步驟 4 中所找到目的地 Outpost 的本機閘道 ID。
9. 選擇 Save changes (儲存變更)。
10. 確定路由的狀態為作用中。

連接來源和目的地存取點的 CIDR 網路範圍之後，您必須建立 AWS Identity and Access Management (IAM) 角色。

建立 IAM 角色

根據預設，所有 S3 on Outposts 資源 (儲存貯體、物件與相關子資源) 皆為私有，且只有資源擁有者才可存取該資源。S3 on Outposts 需要從來源 Outposts 儲存貯體讀取和複寫物件的許可。您可建立 IAM 服務角色並在您的複寫組態中指定此角色以授予這些許可。

本節說明信任政策與最低必要許可政策。這些範例演練提供建立 IAM 角色的逐步說明。如需詳細資訊，請參閱建立 Outposts 上的複寫規則。如需 IAM 角色的詳細資訊，請參閱 IAM 使用者指南中的 IAM 角色。

- 以下範例顯示信任政策，您可以在其中將 S3 on Outposts 識別為可擔任該角色的服務主體。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3-outposts.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

- 以下範例顯示存取政策，您可以在其中授予角色許可來代您執行複寫作業。當 S3 on Outposts 擔任該角色時，即具備您在此政策中指定的許可。若要使用此政策，請以您自己的資訊取代 *user input placeholders*。請務必將其取代為來源和目的地 Outposts 的 Outpost ID，以及來源和目的地 Outposts 儲存貯體的儲存貯體名稱和存取點名稱。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3-outposts:GetObjectVersionForReplication",
            "s3-outposts:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
bucket/SOURCE-OUTPOSTS-BUCKET/object/*",
            "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3-outposts:ReplicateObject",
            "s3-outposts:ReplicateDelete"
         ],
         "Resource":[
```

```
            "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/
bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
            "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/
accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
         ]
      }
   ]
}
```

存取政策會授予下列動作的許可：

- `s3-outposts:GetObjectVersionForReplication` — 對所有物件授予此動作的許可，以允許 S3 on Outposts 取得與每個物件相關聯的特定物件版本。

- `s3-outposts:GetObjectVersionTagging` – *SOURCE-OUTPOSTS-BUCKET* 儲存貯體 (來源儲存貯體) 中物件上這個動作的許可，其允許 S3 on Outposts 讀取要複寫的物件標籤。如需詳細資訊，請參閱新增 S3 on Outposts 儲存貯體的標籤。如果 S3 on Outposts 不具備這些許可，則會複寫物件，但不會複寫物件標籤。
- `s3-outposts:ReplicateObject` 與 `s3-outposts:ReplicateDelete` – *DESTINATION-OUTPOSTS-BUCKET* 儲存貯體 (目的地儲存貯體) 中所有物件上這些動作的許可，其允許 S3 on Outposts 將物件或刪除標記複寫至目的地 Outposts 儲存貯體。如需刪除標記的資訊，請參閱 刪除操作對複寫的影響。

  > **Note**
  >
  > - *DESTINATION-OUTPOSTS-BUCKET* 儲存貯體 (目的地儲存貯體) 上 `s3-outposts:ReplicateObject` 動作的許可也允許複寫物件標籤。因此，您不需要明確授予 `s3-outposts:ReplicateTags` 動作的許可。
  > - 對於跨帳戶複寫，目的地 Outposts 儲存貯體的擁有者必須更新其儲存貯體政策，以授予對 *DESTINATION-OUTPOSTS-BUCKET* 的 `s3-outposts:ReplicateObject` 動作許可。`s3-outposts:ReplicateObject` 動作可讓 S3 on Outposts 將物件和物件標籤複寫到目的地 Outposts 儲存貯體。

如需 S3 on Outposts 動作的清單，請參閱 S3 on Outposts 定義的動作。

> **Important**
>
> AWS 帳戶 擁有 IAM 角色的 必須具有授予 IAM 角色之動作的許可。

例如，假設來源 Outposts 儲存貯體包含另一個 AWS 帳戶所擁有的物件。物件的擁有者必須透過儲存貯體政策和存取點政策，明確授予 AWS 帳戶 擁有 IAM 角色的必要許可。否則，S3 on Outposts 就無法存取這些物件，而導致物件的複寫失敗。
此處描述的許可與基本複寫組態相關。如果您選擇新增額外的複寫組態，則必須將額外許可授予給 S3 on Outposts。

當來源和目的地 Outposts 儲存貯體由不同 擁有時授予許可 AWS 帳戶

當來源和目的地 Outposts 儲存貯體不屬於相同帳戶時，目的地 Outposts 儲存貯體的擁有者必須更新目的地儲存貯體的儲存貯體和存取點政策。這些政策必須對來源 Outposts 儲存貯體和 IAM 服務角色的擁有者授予執行複寫動作的許可，如同下列政策範例所示，若未授予，複寫則會失敗。在這些政策範例中，*DESTINATION-OUTPOSTS-BUCKET* 是目的地儲存貯體。若要使用這些政策範例，請以您自己的資訊取代 *user input placeholders*。

如果您要手動建立 IAM 服務角色，請將角色路徑設定為 `role/service-role/`，如下列政策範例所示。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM ARN。

```
{
   "Version":"2012-10-17",
   "Id":"PolicyForDestinationBucket",
   "Statement":[
      {
         "Sid":"Permissions on objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action":[
            "s3-outposts:ReplicateDelete",
            "s3-outposts:ReplicateObject"
         ],
         "Resource":[
            "arn:aws:s3-outposts:region:DestinationBucket-account-
ID:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*"
         ]
      }

   ]
}
```

```
{
"Version":"2012-10-17",
   "Id":"PolicyForDestinationAccessPoint",
   "Statement":[
      {
         "Sid":"Permissions on objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action":[
            "s3-outposts:ReplicateDelete",
            "s3-outposts:ReplicateObject"
         ],
         "Resource" :[
            "arn:aws:s3-outposts:region:DestinationBucket-account-
ID:outpost/DESTINATION-OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/
object/*"
         ]
      }
   ]
}
```

Note

如果來源 Outposts 儲存貯體中的物件已標記，請注意下列情況：
如果來源 Outposts 儲存貯體擁有者將 `s3-outposts:GetObjectVersionTagging` 與 `s3-outposts:ReplicateTags` 動作的許可授予 S3 on Outposts 來複寫物件標籤 (透過 IAM 角色)，Amazon S3 會連同物件一起複寫標籤。如需 IAM 角色的資訊，請參閱 建立 IAM 角色。

建立 Outposts 上的複寫規則

S3 Replication on Outposts 是跨相同或不同儲存貯體的物件自動非同步複寫 AWS Outposts。複寫會將來源 Outposts 儲存貯體中新建立的物件和物件更新複製至目的地 Outposts 儲存貯體。如需詳細資訊，請參閱複寫 S3 on Outposts 的物件。

> **Note**
>
> 不會複寫您設定複寫規則之前就已存在於來源 Outposts 儲存貯體中的物件。換句話說，S3 on Outposts 不會追溯複寫物件。若要複寫您在複寫組態之前建立的物件，您可使用 `CopyObject` API 操作以將其複製到相同的儲存貯體。複製物件之後，這些物件會在儲存貯體中顯示為「新」物件，且複寫組態將套用至這些物件。如需複製物件的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的 使用 在 Amazon S3 on Outposts 儲存貯體中複製物件 適用於 Java 的 AWS SDK 和 CopyObject。

設定複寫時，會將複寫規則新增至來源 Outposts 儲存貯體。複寫規則會定義要複寫的來源 Outposts 儲存貯體物件，以及存放已複寫物件的目的地 Outposts 儲存貯體。您可以建立規則，以特定的金鑰名稱前綴、一或多個物件標籤、或兩種都用，複寫儲存貯體中的所有物件，或一部分的物件。目的地 Outposts 儲存貯體可以在與來源 Outposts 儲存貯體相同的 Outpost 中，也可以在不同的 Outpost 中。

對於 S3 on Outposts 複寫規則，您必須同時提供來源 Outposts 儲存貯體的存取點 Amazon Resource Name (ARN) 和目的地 Outposts 儲存貯體的存取點 ARN，而不是來源和目的地 Outposts 儲存貯體名稱。

如果您指定要刪除的物件版本 ID，S3 on Outposts 會刪除來源 Outposts 儲存貯體中的該物件版本。但不會在目的地 Outposts 儲存貯體中進行刪除。換句話說，它不會從目的地 Outposts 儲存貯體中刪除相同的物件版本。此行為可防止資料遭到惡意刪除。

當您將複寫規則新增至 Outposts 儲存貯體時，預設會啟用此規則，讓規則在您儲存之後立即運作。

在此範例中，您會設定來源與目的地 Outposts 儲存貯體為不同 Outposts 且同一 AWS 帳戶所擁有的複寫。提供使用 Amazon S3 主控台、 AWS Command Line Interface (AWS CLI) 和 適用於 Java 的 AWS SDK 的範例 適用於 .NET 的 AWS SDK。如需跨帳戶 S3 on Outposts 許可的相關資訊，請參閱 當來源和目的地 Outposts 儲存貯體由不同 擁有時授予許可 AWS 帳戶。

如需設定 S3 on Outposts 複寫規則的先決條件，請參閱 建立複寫規則的先決條件。

使用 S3 主控台

當目的地 Amazon S3 on Outposts 儲存貯體與來源 Outposts 儲存貯體位於不同的 Outpost 時，請依照這些步驟設定複寫規則。

如果目的地 Outposts 儲存貯體位在與來源 Outposts 儲存貯體不同的帳戶中，您必須將儲存貯體政策新增至目的地 Outposts 儲存貯體，以將複寫目的地 Outposts 儲存貯體中物件的許可授予來源 Outposts 儲存貯體帳戶擁有者。

建立複寫規則

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台：//https：/https://console.aws.amazon.com/s3/.microsoft.com。
2. 在 Outposts 儲存貯體清單中，選擇您要使用做為來源儲存貯體的儲存貯體名稱。
3. 選擇管理索引標籤，向下捲動至複寫規則區段，然後選擇建立複寫規則。
4. 對於複寫規則名稱，輸入規則名稱，以利之後識別此規則。此名稱為必要，且在儲存貯體內必須是唯一的。
5. 在狀態下，預設會選擇已啟用。已啟用規則在您儲存它之後就會立即運作。如果希望稍後再啟用此規則，請選擇已停用。
6. 在優先順序下，如果發生規則重疊，則規則的優先順序值會決定要套用的規則。當物件包含在多個複寫規則的範圍內時，S3 on Outposts 會使用這些優先順序值來避免衝突。依預設，新規則會以最高優先順序新增至複製組態。數字愈高，優先順序愈高。

   若要變更規則的優先順序，請在儲存規則之後，從複製規則清單中選擇規則名稱、選擇動作，然後選擇編輯優先順序。
7. 在來源儲存貯體之下，您有下列選項可用來設定複寫來源：

   - 若要複寫整個儲存貯體，請選擇套用至儲存貯體中的所有物件。
   - 若要將字首或標籤篩選套用至複寫來源，請選擇使用一或多個篩選條件限制此規則的範圍。您可以合併字首與標籤。
     - 若要複寫具有相同字首的所有物件，請在字首下的方塊中輸入字首。使用字首篩選條件以限制複寫名稱以相同字串 (例如，`pictures`) 開頭的所有物件。

       如果您輸入的字首是資料夾名稱，您必須使用 / (正斜線) 作為最後一個字元 (例如，`pictures/`)。
     - 若要複寫具有一個或多個相同物件標籤的所有物件，請選擇新增標籤，然後在方塊中輸入鍵/值對。若要新增另一個索引標籤，請重複此程序，。如需物件標籤的詳細資訊，請參閱 新增 S3 on Outposts 儲存貯體的標籤。
8. 若要存取 S3 on Outposts 來源儲存貯體以進行複寫，請在來源存取點名稱下選擇連接至來源儲存貯體的存取點。
9. 在目的地下，選擇您想要 S3 on Outposts 在其中複寫物件的目的地 Outposts 儲存貯體的存取點 ARN。目的地 Outposts 儲存貯體可以 AWS 帳戶 與來源 Outposts 儲存貯體位於相同或不同位置。

如果目的地儲存貯體位在與來源 Outposts 儲存貯體不同的帳戶中，您必須將儲存貯體政策新增至目的地 Outposts 儲存貯體，以將複寫目的地 Outposts 儲存貯體中物件的許可授予來源 Outposts 儲存貯體帳戶擁有者。如需詳細資訊，請參閱[當來源和目的地 Outposts 儲存貯體由不同 擁有時授予許可 AWS 帳戶](#)。

> **Note**
>
> 如果未在目的地 Outposts 儲存貯體上啟用版本控制，您會收到包含啟用版本控制按鈕的警告訊息。選擇此按鈕，以在儲存貯體上啟用版本控制。

10. 設定 S3 on Outposts 可擔任的 AWS Identity and Access Management (IAM) 服務角色，以代表您複寫物件。

    若要設定 IAM 角色，請在 IAM 角色下，執行下列其中一個動作：

    - 若要讓 S3 on Outposts 為您的複寫組態建立新的 IAM 角色，請選擇從現有的 IAM 角色中選擇，然後選擇建立新角色。當您儲存規則時，系統會為符合所選擇來源與目的地 Outposts 儲存貯體的 IAM 角色產生新原則。建議您選擇建立新角色。
    - 您也可以選擇使用現有 IAM 角色。如果這麼做，則必須選擇將必要複寫許可授予 S3 on Outposts 的角色。如果此角色未依您的複寫規則授予 S3 on Outposts 足夠的許可，則複寫會失敗。

      若要選擇現有角色，請選擇從現有 IAM 角色中選擇，然後從下拉式功能表中選擇角色。您也可以選擇輸入 IAM 角色 ARN，然後輸入 IAM 角色的 Amazon Resource Name (ARN)。

    > **Important**
    >
    > 當您新增複寫規則至 S3 on Outposts 儲存貯體時，必須擁有 `iam:CreateRole` 和 `iam:PassRole` 許可，才能建立和傳遞授予 S3 on Outposts 複寫許可的 IAM 角色。如需詳細資訊，請參閱《IAM 使用者指南》中[授予使用者將角色傳遞至 AWS 服務的許可](#)。

11. 依預設，Outposts 儲存貯體中的所有物件都會加密。如需 S3 on Outposts 加密的詳細資訊，請參閱 [S3 on Outposts 中的資料加密](#)。只有使用 Amazon S3 管理金鑰的伺服器端加密 (SSE-S3) 進行加密的物件才可加以複寫。不支援使用 AWS Key Management Service (AWS KMS) 金鑰的伺服器端加密 (SSE-KMS) 或使用經由客戶提供加密金鑰的伺服器端加密 (SSE-C) 進行加密的物件複寫。
12. 設定複寫規則組態時，視需要啟用下列其他選項：

- 如果您想要在複寫組態中啟用 S3 on Outposts 複寫指標，請選取複寫指標。如需詳細資訊，請參閱使用複寫指標監控進度。
- 如果您想要在複寫組態中啟用刪除標記複寫，請選取 Delete marker replication (刪除標記複寫)。如需詳細資訊，請參閱刪除操作對複寫的影響。
- 如果您要將對複本所做的中繼資料變更複寫回來源物件，請選取複本修改同步。如需詳細資訊，請參閱啟用 Outposts 上的 Amazon S3 複本修改同步時的複寫狀態。

13. 若要完成，請選擇建立規則。

儲存規則之後，即可編輯、啟用、停用或刪除規則。若要這麼做，請前往來源 Outposts 儲存貯體的管理索引標籤，向下捲動至複寫規則區段，選擇您的規則，然後選擇編輯規則。

使用 AWS CLI

若要在來源和目的地 Outposts 儲存貯體由相同 擁有時使用 AWS CLI 設定複寫 AWS 帳戶，請執行下列動作：

- 建立來源與目的地 Outposts 儲存貯體。
- 在兩個儲存貯體上啟用版本控制。
- 建立 IAM 角色，授予 S3 on Outposts 複寫物件的許可。
- 將複寫組態新增至來源 Outposts 儲存貯體。

您可以測試以驗證設定。

在來源和目的地 Outposts 儲存貯體由相同 擁有時設定複寫 AWS 帳戶

1. 設定 AWS CLI的憑證描述檔。在此範例中，我們使用描述檔名稱 `acctA`。如需設定憑證描述檔的相關資訊，請參閱《AWS Command Line Interface 使用者指南》中的具名描述檔。

> **Important**
>
> 用於此練習的描述檔必須有必要的許可。例如，您可以在複寫組態中指定 S3 on Outposts 可以擔任的 IAM 服務角色。只有當您所用的描述檔有 `iam:CreateRole` 和 `iam:PassRole` 許可時才可執行此作業。如需詳細資訊，請參閱《IIAM 使用者指南》《中授予使用者將角色傳遞至 AWS 服務的許可。如果您使用管理員憑證建立具名描述檔，具名描述檔將擁有執行所有任務的必要許可。

2. 建立 *source* 儲存貯體並對它啟用版本控制。下列 create-bucket 命令會在美國東部 (維吉尼亞北部) (us-east-1) 區域中建立 *SOURCE-OUTPOSTS-BUCKET* 儲存貯體。若要使用此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control create-bucket --bucket SOURCE-OUTPOSTS-BUCKET --outpost-id SOURCE-OUTPOST-ID --profile acctA --region us-east-1
```

下列 put-bucket-versioning 命令啟用 *SOURCE-OUTPOSTS-BUCKET* 儲存貯體上的版本控制。若要使用此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

3. 建立 *destination* 儲存貯體並對它啟用版本控制。下列 create-bucket 命令會在美國西部 (奧勒岡) (us-west-2) 區域中建立 *DESTINATION-OUTPOSTS-BUCKET* 儲存貯體。若要使用此命令，請以您自己的資訊取代 *user input placeholders*。

> **Note**
>
> 若要在來源和目的地 Outposts 儲存貯體位於相同 時設定複寫組態 AWS 帳戶，請使用相同的具名設定檔。此範例使用 acctA。若要在儲存貯體由不同 擁有時測試複寫組態 AWS 帳戶，請為每個儲存貯體指定不同的設定檔。

```
aws s3control create-bucket --bucket DESTINATION-OUTPOSTS-BUCKET --create-bucket-configuration LocationConstraint=us-west-2 --outpost-id DESTINATION-OUTPOST-ID --profile acctA --region us-west-2
```

下列 put-bucket-versioning 命令啟用 *DESTINATION-OUTPOSTS-BUCKET* 儲存貯體上的版本控制。若要使用此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

4. 建立 IAM 服務角色。稍後在複寫組態中，您會將此服務角色新增至 *SOURCE-OUTPOSTS-BUCKET* 儲存貯體。S3 on Outposts 就會擔任此角色以代您複寫物件。建立 IAM 角色需要兩個步驟：

a. 建立 IAM 角色。

i. 複製下列信任政策，並將它儲存至本機電腦目前目錄下名稱為 s3-on-outposts-role-trust-policy.json 的檔案中。此政策會授予 S3 on Outposts 服務主體擔任該服務角色的許可。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3-outposts.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

ii. 執行下列 命令以建立角色。以您自己的資訊取代 *user input placeholders*。

```
aws iam create-role --role-name replicationRole --assume-role-policy-
document file://s3-on-outposts-role-trust-policy.json --profile acctA
```

b. 將許可政策連接到服務角色。

i. 複製下列許可政策，並將它儲存至本機電腦目前目錄中名為 s3-on-outposts-role-permissions-policy.json 的檔案。此政策會授予各種 S3 on Outposts 儲存貯體與物件動作的許可。若要使用此政策，請以您自己的資訊取代 *user input placeholders*。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3-outposts:GetObjectVersionForReplication",
            "s3-outposts:GetObjectVersionTagging"
         ],
         "Resource":[
```

```
            "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-
OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET/object/*",
            "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-
OUTPOST-ID/accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3-outposts:ReplicateObject",
            "s3-outposts:ReplicateDelete"
         ],
         "Resource":[
            "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-
OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
            "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-
OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
         ]
      }
   ]
}
```

ii. 執行下列命令以建立政策，並將它連接至角色。以您自己的資訊取代 *user input placeholders*。

```
aws iam put-role-policy --role-name replicationRole --policy-
document file://s3-on-outposts-role-permissions-policy.json --policy-
name replicationRolePolicy --profile acctA
```

5. 將複寫組態新增至 *SOURCE-OUTPOSTS-BUCKET* 儲存貯體。

a. 雖然 S3 on Outposts API 需要 XML 格式的複寫組態，但 AWS CLI 需要您以 JSON 格式指定複寫組態。將下列 JSON 儲存至您電腦本機目錄下的 replication.json 檔案中。若要使用此組態，請以您自己的資訊取代 *user input placeholders*。

```
{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": "Tax"},
```

```
      "Destination": {
        "Bucket":
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-
ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT"
      }
    }
  ]
}
```

b. 執行下列 put-bucket-replication 命令，將複寫組態新增至您的來源 Outposts 儲存貯體。若要使用此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control put-bucket-replication --account-id 123456789012 --
bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-
ID/bucket/SOURCE-OUTPOSTS-BUCKET --replication-configuration file://
replication.json --profile acctA
```

c. 使用 get-bucket-replication 命令來擷取複寫組態。若要使用此命令，請以您自己的資訊取代 *user input placeholders*。

```
aws s3control get-bucket-replication --account-id 123456789012 --bucket
 arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
bucket/SOURCE-OUTPOSTS-BUCKET --profile acctA
```

6. 在 Amazon S3 主控台中測試設定：

a. 登入 AWS Management Console 並開啟位於 https：//Amazon S3 主控台。 https://console.aws.amazon.com/s3/

b. 在 *SOURCE-OUTPOSTS-BUCKET* 儲存貯體中，建立名為 Tax 的資料夾。

c. 將範例物件新增至 *SOURCE-OUTPOSTS-BUCKET* 儲存貯體的 Tax 資料夾。

d. 在 *DESTINATION-OUTPOSTS-BUCKET* 儲存貯體中驗證下列事項：

- S3 on Outposts 已複寫物件。

> **Note**
>
> S3 on Outposts 複寫物件所需的時間長短取決於物件大小。如需如何查看複寫狀態的相關資訊，請參閱 取得複寫狀態資訊。

- 在物件屬性標籤中，複寫狀態已設定為複寫 (將此識別為複本物件)。

管理複寫

本節描述 S3 on Outposts 中可用的其他複寫組態選項、如何判斷複寫狀態，以及如何疑難排解複寫。如需核心複寫組態的資訊，請參閱 設定複寫。

主題

- 使用複寫指標監控進度
- 取得複寫狀態資訊
- 故障排除複寫
- 針對 Outposts 上的 S3 複寫使用 EventBridge

使用複寫指標監控進度

Outposts 上的 S3 複寫提供了複寫組態中複寫規則的詳細指標。透過追蹤擱置複寫的位元組、複寫延遲和作業擱置，您可以每 5 分鐘以複寫指標，監控複寫進度。您也可以設定 Amazon EventBridge 以接收複寫失敗通知，協助疑難排解任何組態問題。

啟用複寫指標後，Outposts 上的 S3 複寫會將下列指標發佈至 Amazon CloudWatch：

- 擱置複寫的位元組 — 針對指定的複寫規則，擱置複寫的物件位元組總數。
- 複寫延遲 — 針對指定的複寫規則，複寫目的地儲存貯體位於來源儲存貯體後方的秒數上限。
- 擱置複寫的作業 – 針對指定的複寫規則，擱置複寫的作業數量。作業包含物件、刪除標記和標籤。

> **Note**
>
> Outposts 上的 S3 複寫指標的計費方式與 CloudWatch 自訂指標相同。如需詳細資訊，請參閱 CloudWatch 定價。

取得複寫狀態資訊

複寫狀態可協助您判斷要由 Amazon S3 on Outposts 複寫之物件的目前狀態。來源物件的複寫狀態將會傳回 `PENDING`、`COMPLETED` 或 `FAILED`。複本的複寫狀態將會傳回 `REPLICA`。

複寫狀態概觀

在複寫中，您有一個來源儲存貯體，您可以在其上設定複寫與 S3 on Outposts 複寫物件的目的地儲存貯體。當您從這些儲存貯體請求物件 (使用 `GetObject`) 或物件中繼資料 (使用 `HeadObject`) 時，S3 on Outposts 會在回應中傳回 `x-amz-replication-status` 標頭，如下所示：

- 當您從來源儲存貯體請求物件時，如果請求中的物件符合複寫資格，S3 on Outposts 即會傳回 `x-amz-replication-status` 標頭。

  例如，假設您在複寫組態中指定物件前綴 `TaxDocs`，告知 S3 on Outposts 複寫僅具有金鑰名稱前綴 `TaxDocs` 的物件。系統會複寫您上傳且具有此金鑰名稱前綴 (例如 `TaxDocs/document1.pdf`) 的任何物件。針對具有此金鑰名稱前綴的物件請求，S3 on Outposts 會傳回 `x-amz-replication-status` 標頭，以及代表物件複寫狀態的下列其中一個值：`PENDING`、`COMPLETED` 或 `FAILED`。

  > **Note**
  >
  > 若物件複寫在您上傳完物件後失敗，則您無法重試複寫。您必須再次上傳物件。對於缺少複寫角色許可或儲存貯體許可等問題，物件會轉換成 `FAILED` 狀態。對於暫時性錯誤，例如，如果儲存貯體或 Outpost 無法使用，複寫狀態不會轉換成 `FAILED`，但會保持 `PENDING`。資源恢復線上狀態後，S3 on Outposts 會繼續複寫這些物件。

- 當您從目的地儲存貯體請求物件時，如果您請求中的物件是 S3 on Outposts 建立的複本，S3 on Outposts 會傳回值為 `REPLICA` 的 `x-amz-replication-status` 標頭。

> **Note**
>
> 在從已啟用複寫的來源儲存貯體中刪除物件之前，您應該先檢查物件的複寫狀態，確保已複寫物件。

啟用 Outposts 上的 Amazon S3 複本修改同步時的複寫狀態

當您的複寫規則啟用 S3 on Outposts 複本修改同步時，複本可以報告 `REPLICA` 以外的狀態。如果中繼資料變更正在複寫過程中，則複寫 `x-amz-replication-status` 標頭會傳回 `PENDING`。如果複本修改同步無法複寫中繼資料，則複寫標頭會傳回 `FAILED`。如果中繼資料正確複寫，複寫標頭會傳回值 `REPLICA`。

故障排除複寫

在您設定複寫之後，如果物件複本未出現在目的地 Amazon S3 on Outposts 儲存貯體中，請使用這些故障診斷技巧以識別並修正問題。

- S3 on Outposts 複寫物件所需的時間長短取決於幾個因素，包括來源和目的地 Outposts 的距離，以及物件的大小。

  您可以檢查來源物件的複寫狀態。如果物件複寫狀態為 `PENDING`，表示 S3 on Outposts 尚未完成複寫。如果物件複寫狀態為 `FAILED`，請檢查來源儲存貯體中所設的複寫組態。
- 在來源儲存貯體中的複寫組態中，驗證下列項目：
  - 目的地儲存貯體的存取點 Amazon Resource Name (ARN) 正確。
  - 金鑰名稱前綴正確。例如，如果您設定組態來複寫具有前綴 `Tax` 的物件，則只會複寫具有 `Tax/document1` 或 `Tax/document2` 等金鑰名稱的物件。不會複寫具有金鑰名稱 `document3` 的物件。
  - 狀態為 `Enabled`。
- 確認在任何儲存貯體上均沒有已暫停的版本控制。來源與目的地儲存貯體都必須啟用版本控制。
- 如果目的地儲存貯體由另一個儲存貯體擁有 AWS 帳戶，請確認儲存貯體擁有者在目的地儲存貯體上有儲存貯體政策，允許來源儲存貯體擁有者複寫物件。如需範例，請參閱「當來源和目的地 Outposts 儲存貯體由不同 擁有時授予許可 AWS 帳戶」。
- 如果目的地儲存貯體中未出現物件複本，可能是下列問題阻礙了複寫作業：
  - 如果來源儲存貯體中的物件是由另一個複寫組態所建立的複本，則 S3 on Outposts 不會複寫該複本。例如，如果您將複寫組態從儲存貯體 A 設定到儲存貯體 B，再設定到儲存貯體 C，則 S3 on Outposts 不會將儲存貯體 B 中的物件複本複寫至儲存貯體 C。

    如果您想要將儲存貯體 A 的物件複製到儲存貯體 B 和儲存貯體 C，請在不同的複寫規則中，為來源儲存貯體複寫組態設定多個儲存貯體目的地。例如，在來源儲存貯體 A 上建立兩個複寫規則，其中一個規則可複寫到目的地儲存貯體 B，另一個規則複寫到目的地儲存貯體 C。
  - 來源儲存貯體擁有者可以授予其他 AWS 帳戶 許可來上傳物件。根據預設，來源儲存貯體擁有者不具其他帳戶所建立之物件的任何許可。複寫組態只會複寫來源儲存貯體擁有者具備存取許可的物件。為了避免複寫問題，來源儲存貯體擁有者可以授予其他 AWS 帳戶 許可，以有條件地建立物件，並需要這些物件的明確存取許可。
- 假設您在複寫組態中新增一個規則，以複寫含特定標籤的物件子集。在此情況下，您必須於建立物件時指派特定標籤金鑰與值，以便 S3 on Outposts 複寫物件。如果您先建立物件，之後才將標籤新增至現有物件，S3 on Outposts 就不會複寫該物件。
- 如果儲存貯體政策拒絕存取下列任何動作的複寫角色，則複寫會失敗：

來源儲存貯體：

```
"s3-outposts:GetObjectVersionForReplication",
"s3-outposts:GetObjectVersionTagging"
```

目的地儲存貯體：

```
"s3-outposts:ReplicateObject",
"s3-outposts:ReplicateDelete",
"s3-outposts:ReplicateTags"
```

- 當物件未複寫至其目的地 Outposts 時，Amazon EventBridge 可以通知您。如需詳細資訊，請參閱針對 Outposts 上的 S3 複寫使用 EventBridge。

針對 Outposts 上的 S3 複寫使用 EventBridge

Amazon S3 on Outposts 會與 Amazon EventBridge 整合，並使用 `s3-outposts` 命名空間。EventBridge 是無伺服器事件匯流排服務，可讓您用於將應用程式與來自各種來源的資料互相連線。如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的什麼是 Amazon EventBridge？

您也可以設定 Amazon EventBridge 以接收複寫失敗事件通知，協助疑難排解任何複寫組態問題。當物件未複寫至其目的地 Outposts 時，EventBridge 可以在執行個體中通知您。如需詳細了解複寫物件的目前狀態，請參閱 複寫狀態概觀。

每當 Outposts 儲存貯體發生事件，S3 on Outposts 就會將事件傳送至 EventBridge。與其他目的地不同，您不需要選取想要傳遞的事件類型。您也可以使用 EventBridge 規則將事件路由至其他目標。啟用 EventBridge 後，S3 on Outposts 會將下列所有事件傳送至 EventBridge。

事件類型	描述	命名空間
`Operation FailedReplication`	複寫規則內的物件複寫失敗。如需詳細了解 Outposts 上的 S3 複寫失敗原因，請參閱 使用 EventBridge 檢視 Outposts 上的 S3 複寫失敗原因。	`s3-outposts`

使用 EventBridge 檢視 Outposts 上的 S3 複寫失敗原因

下表列出 Outposts 上的 S3 複寫失敗原因。您可以將 EventBridge 規則設定為透過 Amazon Simple Queue Service (Amazon SQS) AWS Lambda、Amazon Simple Notification Service (Amazon SNS) 或 Amazon CloudWatch Logs 發佈和檢視失敗原因。如需詳細了解針對 EventBridge 使用這些資源的必要權限，請參閱針對 EventBridge 使用資源型政策。

複寫失敗原因	描述
`AssumeRoleNotPermitted`	S3 on Outposts 無法擔任複寫組態中指定的 AWS Identity and Access Management (IAM) 角色。
`DstBucketNotFound`	S3 on Outposts 找不到複寫組態中指定的目的地儲存貯體。
`DstBucketUnversioned`	Outposts 目的地儲存貯體上未啟用版本控制。若要以 Outposts 上的 S3 複寫來複寫物件，您必須啟用目的地儲存貯體上的版本控制。
`DstDelObjNotPermitted`	S3 on Outposts 無法將刪除項目複寫到目的地儲存貯體。可能缺少目的地儲存貯體的 `s3-outposts:ReplicateDelete` 許可。
`DstMultipartCompleteNotPermitted`	S3 on Outposts 無法完成目的地儲存貯體中物件的分段上傳。可能缺少目的地儲存貯體的 `s3-outposts:ReplicateObject` 許可。
`DstMultipartInitNotPermitted`	S3 on Outposts 無法起始目的地儲存貯體中物件的分段上傳。可能缺少目的地儲存貯體的 `s3-outposts:ReplicateObject` 許可。
`DstMultipartPartUploadNotPermitted`	S3 on Outposts 無法在目的地儲存貯體上傳分段物件。可能缺少目的地儲存貯體的 `s3-outposts:ReplicateObject` 許可。

複寫失敗原因	描述
DstOutOfCapacity	S3 on Outposts 無法複寫到目的地 Outpost，因為 Outpost 不在 S3 儲存容量中。
DstPutObjNotPermitted	S3 on Outposts 無法將物件複寫到目的地儲存貯體。可能缺少目的地儲存貯體的 s3-outposts:ReplicateObject 許可。
DstPutTaggingNotPermitted	S3 on Outposts 無法將物件標籤複寫到目的地儲存貯體。可能缺少目的地儲存貯體的 s3-outposts:ReplicateObject 許可。
DstVersionNotFound	S3 on Outposts 無法在目的地儲存貯體中找到所需的物件版本，以複寫該物件版本的中繼資料。
SrcBucketReplicationConfigMissing	S3 on Outposts 找不到與來源 Outposts 儲存貯體相關聯的存取點複寫組態。
SrcGetObjNotPermitted	S3 on Outposts 無法存取來源儲存貯體中的物件以進行複寫。可能缺少來源儲存貯體的 s3-outposts:GetObjectVersionForReplication 許可。
SrcGetTaggingNotPermitted	S3 on Outposts 無法從來源儲存貯體存取物件標籤資訊。可能缺少來源儲存貯體的 s3-outposts:GetObjectVersionTagging 許可。
SrcHeadObjectNotPermitted	S3 on Outposts 無法從來源儲存貯體擷取物件中繼資料。可能缺少來源儲存貯體的 s3-outposts:GetObjectVersionForReplication 許可。
SrcObjectNotEligible	物件不符合複寫資格。物件或物件標籤不符合複寫組態。

如需詳細了解複寫疑難排解，請參閱下列主題：

- 建立 IAM 角色
- 故障排除複寫

以 CloudWatch 監控 EventBridge

Amazon EventBridge 與 Amazon CloudWatch 整合用以進行監控。EventBridge 會每分鐘自動將指標傳送至 CloudWatch。這些指標包括已符合規則的事件數量，以及規則叫用目標的次數。在 EventBridge 中執行規則時，與該規則關聯的所有目標都會受到叫用。您可採取下列方式，透過 CloudWatch 監控 EventBridge 行為。

- 您可以從 CloudWatch 儀表板，監控 EventBridge 規則的可用 EventBridge 指標。然後，您可以使用 CloudWatch 的功能 (例如 CloudWatch 警示) 在特定指標上設定警示。如果這些指標達到您在警示中指定的自訂閾值，您就會收到通知，且可採取相應動作。
- 您可以將 Amazon CloudWatch Logs 設定為 EventBridge 規則的目標。接著，EventBridge 會建立日誌串流，而 CloudWatch Logs 會將事件中的文字儲存為日誌項目。如需詳細資訊，請參閱 EventBridge 和 CloudWatch Logs。

如需詳細了解偵錯 EventBridge 事件傳遞和封存事件，請參閱下列主題：

- 事件重試政策和使用無效字母佇列
- 封存 EventBridge 事件

使用 共用 S3 on Outposts AWS RAM

Amazon S3 on Outposts 支援使用 AWS Resource Access Manager () 跨組織內的多個帳戶共用 S3 容量AWS RAM。透過 S3 on Outposts 共用，您可以允許其他人在您的 Outpost 上建立和管理儲存貯體、端點和存取點。

本主題示範如何使用 AWS RAM 與 AWS 組織中 AWS 帳戶 的另一個 共用 S3 on Outposts 和相關資源。

先決條件

- Outpost 擁有者帳戶已在 AWS Organizations中設定組織。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的建立組織。

- 組織包含您要與 AWS 帳戶 之共用 S3 on Outposts 容量的 。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的傳送邀請給 AWS 帳戶。
- 選取您要共用的下列選項之一。必須選取第二個資源 (子網路或 Outposts)，才能存取端點。端點是要存取存放在 S3 on Outposts 上的資料時的一項網路要求。

選項 1	選項 2
S3 on Outposts 允許使用者在 Outposts 和存取點上建立儲存貯體，並將物件新增到這些儲存貯體。	S3 on Outposts 允許使用者在 Outposts 和存取點上建立儲存貯體，並將物件新增到這些儲存貯體。
子網路 允許使用者使用您的 Virtual Private Cloud (VPC) 以及與您的子網路關聯的端點。	Outposts 允許使用者查看 S3 容量圖表和 AWS Outposts 主控台首頁。也允許使用者在共用 Outposts 上建立子網路並建立端點。

程序

1. AWS Management Console 使用 AWS 帳戶 擁有 Outpost 的 登入 ，然後開啟位於 https：//https://console.aws.amazon.com/ram/home 的 AWS RAM 主控台。
2. 請確定您已在 AWS Organizations 中啟用與 共用 AWS RAM。如需詳細資訊，請參閱《AWS RAM 使用者指南》中的在 AWS Organizations中啟用資源共用。
3. 使用先決條件中的選項 1 或選項 2 建立資源共享。如果您有多個 S3 on Outposts 資源，請選取要共用的資源的 Amazon Resource Name (ARN)。若要啟用端點，請共用您的子網路或 Outpost。

   如需如何建立資源共用的資訊，請參閱《AWS RAM 使用者指南》中的建立資源共用。
4. 您共用資源 AWS 帳戶 的 現在應該能夠使用 S3 on Outposts。根據您在先決條件中選取的選項，請向帳戶使用者提供下列資訊：

選項 1	選項 2
Outpost ID VPC ID	Outpost ID

選項 1	選項 2
子網路 ID	
安全群組 ID	

> **Note**
>
> 使用者可以使用 AWS RAM 主控台、 AWS Command Line Interface (AWS CLI)、 AWS SDKs或 REST API 來確認資源已與其共用。使用者可以使用 get-resource-shares CLI 命令來查看其現有資源共用。

使用範例

在您與另一個帳戶共用您的 S3 on Outposts 資源後，該帳戶可以管理您的 Outpost 上的儲存貯體和物件。如果您共用 Subnets (子網路) 資源，那麼該帳戶可以使用您建立的端點。下列範例示範使用者如何在共用這些資源後 AWS CLI ，使用 與您的 Outpost 互動。

Example ：建立儲存貯體

下列範例會在 Outpost *op-01ac5d28a6a232904* 上建立名為 *amzn-s3-demo-bucket1* 的儲存貯體。在使用此命令之前，請將每個 *user input placeholder* 取代為適合您的使用案例的值。

```
aws s3control create-bucket --bucket amzn-s3-demo-bucket1 --outpost-
id op-01ac5d28a6a232904
```

如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 create-bucket。

Example ：建立存取點

下列範例會使用下表中的範例參數，在 Outpost 上建立存取點。使用此命令之前，請將這些*user input placeholder*值和 AWS 區域 程式碼取代為您的使用案例的適當值。

Parameter (參數)	Value
帳戶 ID	*111122223333*

Parameter (參數)	Value
存取點名稱	*example-outpost-access-point*
Outpost ID	*op-01ac5d28a6a232904*
Outpost 儲存貯體名稱	*amzn-s3-demo-bucket1*
VPC ID	*vpc-1a2b3c4d5e6f7g8h9*

> **Note**
>
> 帳戶 ID 參數必須是儲存貯體擁有者的 AWS 帳戶 ID，即共用使用者。

```
aws s3control create-access-point --account-id 111122223333 --name example-outpost-
access-point \
--bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/
bucket/amzn-s3-demo-bucket1 \
--vpc-configuration VpcId=vpc-1a2b3c4d5e6f7g8h9
```

如需此命令的相關詳細資訊，請參閱 AWS CLI 參考中的 create-access-point。

Example ：上傳物件

下列範例會透過由 AWS 帳戶 *111122223333* 擁有之 Outpost *op-01ac5d28a6a232904* 上的存取點 *example-outpost-access-point*，從使用者的本機檔案系統上傳檔案 *my_image.jpg* 到名為 *images/my_image.jpg* 的物件。使用此命令之前，請將這些*user input placeholder*值和 AWS 區域 程式碼取代為您的使用案例的適當值。

```
aws s3api put-object --bucket arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-
point \
--body my_image.jpg --key images/my_image.jpg
```

如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 put-object。

> **Note**
>
> 如果此操作導致 Resource not found (找不到資源) 錯誤或無回應，則您的 VPC 可能沒有共用端點。
> 若要檢查是否有共用端點，請使用 list-shared-endpoints AWS CLI 命令。如果沒有共享端點，請與 Outpost 擁有者合作建立一個端點。如需詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的 ListSharedEndpoints。

Example ：建立端點

下列範例在共用 Outpost 上建立端點。在使用此命令之前，請將 Outpost ID、子網路 ID 和安全群組 ID 的 *user input placeholder* 值取代為適合您的使用案例的值。

> **Note**
>
> 使用者只有在資源共用包含 Outposts 資源時，才能執行此操作。

```
aws s3outposts create-endpoint --outposts-id op-01ac5d28a6a232904 --subnet-id XXXXXX --
security-group-id XXXXXXX
```

如需此命令的詳細資訊，請參閱 AWS CLI 參考中的 create-endpoint。

使用 S3 on Outposts AWS 服務 的其他

其他在 本機執行 AWS 服務 的 AWS Outposts 也可以使用您的 Amazon S3 on Outposts 容量。在 Amazon CloudWatch 中，`S3Outposts` 命名空間顯示 S3 on Outposts 內儲存貯體指標的詳細資訊，但這些指標不包括其他 AWS 服務的使用量。若要管理其他人使用的 S3 on Outposts 容量 AWS 服務，請參閱下表中的資訊。

AWS 服務	描述	進一步了解
Amazon S3	所有 S3 on Outposts 使用量都有匹配的帳戶和儲存貯體 CloudWatch 指標。	請參閱指標

AWS 服務	描述	進一步了解
Amazon Elastic Block Store (Amazon EBS)	對於 Amazon EBS on Outposts，您可以選擇 AWS Outpost 作為快照目的地，並在本機存放在 S3 on Outpost 中。	進一步了解
Amazon Relational Database Service (Amazon RDS)	您可以使用 Amazon RDS 本地備份將 RDS 備份存放在本地 Outpost 上。	進一步了解

監控 S3 on Outposts

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (OUTPOSTS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

如需監控 Amazon S3 on Outposts 儲存貯體容量的詳細資訊，請參閱下列主題。

主題

- 使用 Amazon CloudWatch 指標管理 S3 on Outposts 容量
- 使用 Amazon CloudWatch Events 接收 S3 on Outposts 事件通知
- 使用 AWS CloudTrail 日誌監控 S3 on Outposts

使用 Amazon CloudWatch 指標管理 S3 on Outposts 容量

若要協助管理 Outposts 上的 S3 容量固定，建議您建立 CloudWatch 提醒，在儲存空間使用率超過特定閾值時告知您。如需 S3 on Outposts 之 CloudWatch 指標的詳細資訊，請參閱 CloudWatch 指標。如果無足夠的空間可以在您的 Outpost 上存放物件，API 會傳回容量不足例外狀況 (ICE)。若要釋放空間，您可以建立 CloudWatch 提醒，其會觸發明確的資料刪除，或使用生命週期到期政策使物件過期。若要在刪除之前儲存資料，您可以使用 AWS DataSync 將資料從 Amazon S3 on Outposts 儲存貯體複製到 中的 S3 儲存貯體 AWS 區域。如需有關使用 DataSync 的詳細資訊，請參閱 AWS DataSync 使用者指南中的 AWS DataSync入門。

CloudWatch 指標

`S3Outposts` 命名空間包含下列 Amazon S3 on Outposts 儲存貯體指標。您可以監控佈建的 S3 on Outposts 位元組總數、物件可用的位元組總數，以及特定儲存貯體的所有物件大小總計。所有直接 S3 用量都存在儲存貯體或帳戶相關指標。間接 S3 用量 (例如在 Outpost 上存放 Amazon Elastic Block Store 本機快照或 Amazon Relational Database Service 備份) 會耗用 S3 容量，但不包含在儲存貯體或帳戶相關指標中。如需 Amazon EBS 本機快照的詳細資訊，請參閱 Outposts 上的 Amazon

EBS 本機快照。若要查看您的 Amazon EBS 成本報告，請造訪 https://console.aws.amazon.com/costmanagement/。

> **Note**
>
> S3 on Outposts 僅支援下列指標，而不支援其他 Amazon S3 指標。
> 由於 S3 on Outposts 具有固定容量限制，因此建議您建立 CloudWatch 提醒，以在儲存空間使用率超過特定閾值時通知您。

指標	描述	時段	單位	Type
Outpost talByte:	Outpost 的佈建容量總計 (位元組)。	5 分鐘	位元組	S3 on Outposts
Outpost eeBytes	存放客戶資料的 Outpost 可用位元組計數。	5 分鐘	位元組	S3 on Outposts
BucketU dBytes	指定儲存貯體的所有物件總大小。	5 分鐘	位元組	S3 on Outposts。僅限直接 S3 用量。
Accountl edBytes	指定 Outposts 帳戶的所有物件大小總計。	5 分鐘	位元組	S3 on Outposts。僅限直接 S3 用量。
BytesPe ingRepl ation	針對指定複寫規則等待複寫的物件位元組總數。如需如何啟用複寫指標的詳細資訊，請參閱在 Outposts 之間建立複寫規則。	5 分鐘	位元組	選用。適用於 S3 Replication on Outposts。
Operati sPendin eplicat n	針對指定複寫規則等待複寫的操作總數。如需如何啟用複寫指標的詳細資訊，請參閱在 Outposts 之間建立複寫規則。	5 分鐘	計數	選用。適用於 S3 Replication on Outposts。
Replica onLaten	針對指定的複寫規則，複寫目的地儲存貯體落後來源儲存貯體的目前延遲秒數。如需如何	5 分鐘	秒鐘	選用。適用於 S3 Replication on Outposts。

指標	描述	時段	單位	Type
	啟用複寫指標的詳細資訊，請參閱在 Outposts 之間建立複寫規則。			

使用 Amazon CloudWatch Events 接收 S3 on Outposts 事件通知

您可以使用 CloudWatch Events 為任何 Amazon S3 on Outposts API 事件建立規則。建立規則時，您可以選擇透過所有受支援的 CloudWatch 目標接收通知，包括 Amazon Simple Queue Service (Amazon SQS)、Amazon Simple Notification Service (Amazon SNS) 和 AWS Lambda。如需詳細資訊，請參閱 Amazon CloudWatch Events 使用者指南中的可作為 CloudWatch Events 目標的AWS 服務清單。若要選擇要使用 S3 on Outposts 的目標服務，請參閱《Amazon CloudWatch Events 使用者指南》中的建立 AWS 在 API 呼叫上使用 觸發的 CloudWatch Events 規則 AWS CloudTrail。 Amazon CloudWatch

> **Note**
>
> 對於 S3 on Outposts 物件操作，只有在您的線索 （選擇性使用事件選取器） 設定為接收這些事件時，CloudTrail 傳送的 AWS API 呼叫事件才會符合您的規則。如需詳細資訊，請參閱 AWS CloudTrail 使用者指南中的使用 CloudTrail 日誌檔案。

Example

以下是 `DeleteObject` 操作的範例規則。若要使用此範例規則，請使用 S3 on Outposts 儲存貯體的名稱取代 *`amzn-s3-demo-bucket1`*。

```
{
  "source": [
    "aws.s3-outposts"
  ],
  "detail-type": [
    "AWS API call through CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3-outposts.amazonaws.com"
    ],
```

```
    "eventName": [
      "DeleteObject"
    ],
    "requestParameters": {
      "bucketName": [
        "amzn-s3-demo-bucket1"
      ]
    }
  }
}
```

使用 AWS CloudTrail 日誌監控 S3 on Outposts

Amazon S3 on Outposts 已與 服務整合 AWS CloudTrail，此服務提供由使用者、角色或 S3 on Outposts AWS 服務 中 所採取動作的記錄。您可以使用 AWS CloudTrail 來取得 S3 on Outposts 儲存貯體層級和物件層級請求的相關資訊，以稽核和記錄 S3 on Outposts 事件活動。

若要針對所有 Outposts 儲存貯體或特定 Outposts 儲存貯體清單啟用 CloudTrail 資料事件，您必須在 CloudTrail 中手動建立追蹤。如需 CloudTrail 日誌檔項目的詳細資訊，請參閱 S3 on Outposts 日誌檔項目。

如需 S3 on Outposts 的 CloudTrail 資料事件完整清單，請參閱《Amazon S3 使用者指南》中的 CloudTrail 中的 Amazon S3 資料事件。

> **Note**
>
> - 最佳實務是為 AWS CloudTrail 資料事件 Outposts 儲存貯體建立生命週期政策。設定生命週期政策，在您需要稽核日誌檔的時段之後，定期移除這些日誌檔。這麼做可降低 Amazon Athena 針對每個查詢分析的資料量。如需詳細資訊，請參閱建立和管理 Amazon S3 on Outposts 儲存貯體的生命週期組態。
> - 如需有關如何查詢 CloudTrail 日誌的範例，請參閱 AWS 大數據部落格文章《使用 AWS CloudTrail 和 Amazon Athena 來分析安全、合規和營運活動》。

針對 S3 on Outposts 儲存貯體中的物件啟用 CloudTrail 記錄

您可以使用 Amazon S3 主控台來設定 AWS CloudTrail 追蹤，以記錄 Amazon S3 on Outposts 儲存貯體中物件的資料事件。CloudTrail 支援記錄 `GetObject`、`DeleteObject` 與 `PutObject` 等 S3 on Outposts 物件層級 API 操作。這些事件稱為資料事件。

根據預設，CloudTrail 追蹤不會記錄資料事件。不過，您可以設定追蹤來記錄所指定之 S3 on Outposts 儲存貯體的資料事件，或記錄 AWS 帳戶中所有 S3 on Outposts 儲存貯體的資料事件。

CloudTrail 不會在 CloudTrail 事件歷程記錄中填入資料事件。此外，並非所有的 S3 on Outposts 儲存貯體層級 API 操作都會填入 CloudTrail 事件歷史記錄中。如需如何查詢 CloudTrail 日誌的詳細資訊，請參閱 AWS 知識中心上的使用 Amazon CloudWatch Logs 篩選模式和 Amazon Athena 查詢 CloudTrail 日誌。

若要設定追蹤來記錄 S3 on Outposts 儲存貯體的資料事件，您可以使用 AWS CloudTrail 主控台或 Amazon S3 主控台。如果您要設定線索來記錄 中所有 S3 on Outposts 儲存貯體的資料事件 AWS 帳戶，則使用 CloudTrail 主控台會更輕鬆。如需使用 CloudTrail 主控台設定追蹤來記錄 S3 on Outposts 資料事件的相關資訊，請參閱《AWS CloudTrail 使用者指南》中的資料事件。

> **Important**
>
> 資料事件需支付額外的費用。如需詳細資訊，請參閱 AWS CloudTrail 定價。

下列程序示範如何使用 Amazon S3 主控台設定 CloudTrail 追蹤，以記錄 S3 on Outposts 儲存貯體的資料事件。

> **Note**
>
> 建立儲存貯體 AWS 帳戶 的 擁有它，而且是唯一可以設定要傳送 S3 on Outposts 資料事件的 AWS CloudTrail。

針對 S3 on Outposts 儲存貯體中的物件啟用 CloudTrail 資料事件記錄

1. 登入 AWS Management Console ，並在 https：//Amazon S3 主控台：//https：/https://console.aws.amazon.com/s3/.microsoft.com。
2. 在左側導覽窗格中，選擇 Outposts buckets (Outposts 儲存貯體)。
3. 選擇您要使用 CloudTrail 記錄其資料事件的 Outposts 儲存貯體名稱。
4. 選擇 Properties (屬性)。
5. 在 AWS CloudTrail 資料事件區段中，然後選擇在 CloudTrail 中設定。

   AWS CloudTrail 主控台隨即開啟。

您可以建立新的 CloudTrail 追蹤或重複使用現有的追蹤，並設定要在追蹤中記錄的 S3 on Outposts 資料事件。

6. 在 CloudTrail 主控台儀表板頁面上，選擇建立追蹤。
7. 在步驟 1 選擇追蹤屬性頁面上，提供追蹤的名稱、選擇 S3 儲存貯體來存放追蹤記錄、指定您想要的任何其他設定，然後選擇下一步。
8. 在步驟 2 選擇日誌事件頁面的事件類型下，選擇資料事件。

   針對資料事件類型，選擇 S3 Outposts。選擇 Next (下一步)。

   > **Note**
   >
   > - 當您建立追蹤，並針對 S3 on Outposts 並設定資料事件記錄時，必須正確地指定資料事件類型。
   >   - 如果您使用 CloudTrail 主控台，請針對資料事件類型選擇 S3 Outposts。如需如何在 CloudTrail 主控台中建立追蹤的相關資訊，請參閱《AWS CloudTrail 使用者指南》中的使用主控台建立和更新追蹤。如需如何在 CloudTrail 主控台中設定 S3 on Outposts 資料事件記錄的相關資訊，請參閱《AWS CloudTrail 使用者指南》中的記錄 Amazon S3 物件的資料事件。
   >   - 如果您使用 AWS Command Line Interface (AWS CLI) 或 AWS SDKs，請將 `resources.type` 欄位設定為 `AWS::S3Outposts::Object`。如需如何使用 記錄 S3 on Outposts 資料事件的詳細資訊 AWS CLI，請參閱AWS CloudTrail 《 使用者指南》中的日誌 S3 on Outposts 事件。
   > - 如果您使用 CloudTrail 主控台或 Amazon S3 主控台設定追蹤，來記錄 S3 on Outposts 儲存貯體的資料事件，則 Amazon S3 主控台會顯示儲存貯體已啟用物件層級記錄。

9. 在步驟 3 檢閱並建立頁面上，檢閱您設定的追蹤屬性和日誌事件。然後，選擇建立追蹤。

針對 S3 on Outposts 儲存貯體中的物件停用 CloudTrail 資料事件記錄

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudtrail/ 開啟 CloudTrail 主控台。
2. 在左側導覽窗格中，選擇追蹤。
3. 選擇您已建立來記錄 S3 on Outposts 儲存貯體之事件的追蹤名稱。
4. 在追蹤的詳細資訊窗格上，選擇右上角的停止記錄。

5. 在出現的對話方塊中，選擇停止記錄。

Amazon S3 on Outposts AWS CloudTrail 日誌檔案項目

Amazon S3 on Outposts 管理事件可透過 取得 AWS CloudTrail。此外，您可以選擇為 AWS CloudTrail中的資料事件啟用日誌記錄功能。

追蹤是一種組態，能讓事件以日誌檔案的形式交付至您所指定區域的 S3 儲存貯體中。Outposts 儲存貯體的 CloudTrail 日誌包含一個新欄位 `edgeDeviceDetails`，可識別指定儲存貯體所在的 Outpost。

其他日誌欄位包含要求的動作、動作的日期和時間，以及要求參數。CloudTrail 日誌檔案並非依公有 API 呼叫追蹤記錄的堆疊排序，因此不會以任何特定順序出現。

以下範例顯示的是展示對 `s3-outposts` 採取 PutObject 動作的 CloudTrail 日誌項目。

```
{
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam::111122223333:user/yourUserName",
        "accountId": "222222222222",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "yourUserName"
      },
      "eventTime": "2020-11-30T15:44:33Z",
      "eventSource": "s3-outposts.amazonaws.com",
      "eventName": "PutObject",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "26.29.66.20",
      "userAgent": "aws-cli/1.18.39 Python/3.4.10 Darwin/18.7.0 botocore/1.15.39",
      "requestParameters": {
        "expires": "Wed, 21 Oct 2020 07:28:00 GMT",
        "Content-Language": "english",
        "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
        "ObjectCannedACL": "BucketOwnerFullControl",
        "x-amz-server-side-encryption": "Aes256",
        "Content-Encoding": "gzip",
        "Content-Length": "10",
        "Cache-Control": "no-cache",
```

```
        "Content-Type": "text/html; charset=UTF-8",
        "Content-Disposition": "attachment",
        "Content-MD5": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
        "x-amz-storage-class": "Outposts",
        "x-amz-server-side-encryption-customer-algorithm": "Aes256",
        "bucketName": "amzn-s3-demo-bucket1",
        "Key": "path/upload.sh"
      },
      "responseElements": {
        "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
        "x-amz-server-side-encryption": "Aes256",
        "x-amz-version-id": "001",
        "x-amz-server-side-encryption-customer-algorithm": "Aes256",
        "ETag": "d41d8cd98f00b204e9800998ecf8427f"
      },
      "additionalEventData": {
        "CipherSuite": "ECDHE-RSA-AES128-SHA",
        "bytesTransferredIn": 10,
        "x-amz-id-2": "29xXQBV2O
+xOHKItvzY1suLv1i6A52E0zOX159fpfsItYd58JhXwKxXAXI4IQkp6",
        "SignatureVersion": "SigV4",
        "bytesTransferredOut": 20,
        "AuthenticationMethod": "AuthHeader"
      },
      "requestID": "8E96D972160306FA",
      "eventID": "ee3b4e0c-ab12-459b-9998-0a5a6f2e4015",
      "readOnly": false,
      "resources": [
        {
          "accountId": "222222222222",
          "type": "AWS::S3Outposts::Object",
          "ARN": "arn:aws:s3-outposts:us-east-1:YYY:outpost/op-01ac5d28a6a232904/
bucket/path/upload.sh"
        },
        {
          "accountId": "222222222222",
          "type": "AWS::S3Outposts::Bucket",
          "ARN": "arn:aws:s3-outposts:us-east-1:YYY:outpost/op-01ac5d28a6a232904/
bucket/"
        }
      ],
      "eventType": "AwsApiCall",
      "managementEvent": false,
```

```
    "recipientAccountId": "444455556666",
    "sharedEventID": "02759a4c-c040-4758-b84b-7cbaaf17747a",
    "edgeDeviceDetails": {
      "type": "outposts",
      "deviceId": "op-01ac5d28a6a232904"
    },
    "eventCategory": "Data"
  }
```

使用 Amazon S3 on Outposts 進行開發

使用 Amazon S3 on Outposts，您可以在 AWS Outposts 上建立 S3 儲存貯體，並針對需要本機資料存取、本機資料處理和資料駐留的應用程式，在內部部署輕鬆存放和擷取物件。S3 on Outposts 提供新的儲存類別 S3 Outposts (OUTPOSTS)，其使用 Amazon S3 APIs，旨在以持久且備援的方式將資料存放在 上的多個裝置和伺服器上 AWS Outposts。您可以使用存取點和透過 Virtual Private Cloud (VPC) 的端點連線，與您的 Outpost 儲存貯體進行通訊。就像在 Amazon S3 儲存貯體一樣，您在 Outpost 儲存貯體上可以使用同樣的 API 和功能，包括存取政策、加密和標記。您可以透過、 AWS Command Line Interface (AWS CLI) AWS Management Console、 AWS SDKs 或 REST API 使用 S3 on Outposts。如需詳細資訊，請參閱 什麼是 Amazon S3 on Outposts？

下列主題提供使用 S3 on Outposts 進行開發的資訊

主題

S3 on Outposts 支援的區域

以下支援 S3 on Outposts AWS 區域。

- 美國東部 (維吉尼亞北部) (us-east-1)
- 美國東部 (俄亥俄) (us-east-2)
- 美國西部 (加利佛尼亞北部) (us-west-1)
- 美國西部 (奧勒岡) (us-west-2)
- 非洲 (開普敦) (af-south-1)
- 亞太區域 （ 雅加達 ） (ap-sourtheast-3)
- 亞太區域 (孟買) (ap-south-1)
- 亞太區域 (大阪) (ap-northeast-3)
- 亞太區域 (首爾) (ap-northeast-2)
- 亞太區域 (新加坡) (ap-southeast-1)
- 亞太區域 (雪梨) (ap-southeast-2)

- 亞太區域 (東京) (ap-northeast-1)
- 加拿大 (中部) (ca-central-1)
- 歐洲 (法蘭克福) (eu-central-1)
- 歐洲 (愛爾蘭) (eu-west-1)
- 歐洲 (倫敦) (eu-west-2)
- 歐洲 (米蘭) (eu-south-1)
- 歐洲 (巴黎) (eu-west-3)
- 歐洲 (斯德哥爾摩) (eu-north-1)
- 以色列 (特拉維夫) (il-central-1)
- 中東 (巴林) (me-south-1)
- 南美洲 (聖保羅) (sa-east-1)
- AWS GovCloud （美國東部）(us-gov-east-1)
- AWS GovCloud （美國西部）(us-gov-west-1)

Amazon S3 on Outposts API 操作

本主題列出了您可以與 Amazon S3 on Outposts 搭配使用的 Amazon S3、Amazon S3 Control 和 Amazon S3 on Outposts API 操作。

主題

適用於管理物件的 Amazon S3 API 操作

S3 on Outposts 設計為與 Amazon S3 一樣使用相同的物件 API 操作。您必須使用存取點來存取 Outpost 儲存貯體中的任何物件。當搭配 S3 on Outposts 使用物件 API 操作時，您可以提供 Outposts 存取點 Amazon Resource Name (ARN) 或存取點別名。如需存取點別名的詳細資訊，請參閱 針對您的 S3 on Outposts 儲存貯體存取點使用儲存貯體樣式別名。

Amazon S3 on Outposts 支援下列 Amazon S3 API 操作：

- AbortMultipartUpload

- CompleteMultipartUpload
- CopyObject
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- DeleteObjectTagging
- GetObject
- GetObjectTagging
- HeadBucket
- HeadObject
- ListMultipartUploads
- ListObjects
- ListObjectsV2
- ListObjectVersions
- ListParts
- PutObject
- PutObjectTagging
- UploadPart
- UploadPartCopy

適用於管理儲存貯體的 Amazon S3 Control API 操作

S3 on Outposts 支援下列適用於儲存貯體的 Amazon S3 Control API 操作。

- CreateAccessPoint
- CreateBucket
- DeleteAccessPoint
- DeleteAccessPointPolicy
- DeleteBucket
- DeleteBucketLifecycleConfiguration
- DeleteBucketPolicy
- DeleteBucketReplication

- DeleteBucketTagging
- GetAccessPoint
- GetAccessPointPolicy
- GetBucket
- GetBucketLifecycleConfiguration
- GetBucketPolicy
- GetBucketReplication
- GetBucketTagging
- GetBucketVersioning
- ListAccessPoints
- ListRegionalBuckets
- PutAccessPointPolicy
- PutBucketLifecycleConfiguration
- PutBucketPolicy
- PutBucketReplication
- PutBucketTagging
- PutBucketVersioning

適用於管理 Outposts 的 S3 on Outposts API 操作

S3 on Outposts 支援下列適用於管理端點的 Amazon S3 on Outposts API 操作。

- CreateEndpoint
- DeleteEndpoint
- ListEndpoints
- ListOutpostsWithS3
- ListSharedEndpoints

使用適用於 Java 的開發套件設定 S3 on Outposts 的 S3 控制用戶端

下列範例使用 適用於 Java 的 AWS SDK為 Amazon S3 on Outposts 設定 Amazon S3 控制用戶端。若要使用此範例，請以您自己的資訊取代每個 *`user input placeholder`*。

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;

public AWSS3Control createS3ControlClient() {

    String accessKey = AWSAccessKey;
    String secretKey = SecretAccessKey;
    BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey, secretKey);

    return AWSS3ControlClient.builder().enableUseArnRegion()
            .withCredentials(new AWSStaticCredentialsProvider(awsCreds))
            .build();

}
```

透過 IPv6 向 S3 on Outposts 提出請求

Amazon S3 on Outposts 和 S3 on Outposts 雙堆疊端點支援使用 IPv6 或 IPv4 通訊協定，對 S3 on Outposts 儲存貯體提出請求。有了 S3 on Outposts 的 IPv6 支援，您可以使用 S3 on Outposts API 來透過 IPv6 網路存取和操作儲存貯體和控制平面資源。

> **Note**
>
> IPv6 網路不支援 S3 on Outposts 物件動作 (例如 `PutObject` 或 `GetObject`)。

透過 IPv6 網路存取 S3 on Outposts 不另行收費。如需有關 S3 on Outposts 的詳細資訊，請參閱 S3 on Outposts 定價。

主題

- IPv6 入門
- 使用雙堆疊端點來透過 IPv6 網路提出請求
- 在 IAM 原則中使用 IPv6 地址
- 測試 IP 地址相容性
- 搭配 AWS PrivateLink使用 IPv6

- 使用 S3 on Outposts 雙堆疊端點

IPv6 入門

若要透過 IPv6 向 S3 on Outposts 儲存貯體提出請求，您必須使用雙堆疊端點。下節說明如何使用雙堆疊端點透過 IPv6 提出請求。

下列是嘗試透過 IPv6 存取 S3 on Outposts 儲存貯體之前的重要考量：

- 存取儲存貯體的用戶端與網路必須啟用才能使用 IPv6。
- 虛擬託管式與路徑式請求都支援 IPv6 存取。如需詳細資訊，請參閱使用 S3 on Outposts 雙堆疊端點。
- 如果您在 AWS Identity and Access Management (IAM) 使用者或 S3 on Outposts 儲存貯體政策中使用來源 IP 地址篩選，則必須更新政策以包含 IPv6 地址範圍。

> **Note**
>
> 此要求僅適用於 S3 on Outposts 儲存貯體操作，以及跨 IPv6 網路的控制平面資源。IPv6 網路不支援 Amazon S3 on Outposts 物件動作。

- 使用 IPv6 時，伺服器會存取 IPv6 格式的日誌檔案輸出 IP 地址。您必須更新用來剖析 S3 on Outposts 日誌檔案的現有工具、指令碼與軟體，讓這些項目可以剖析 IPv6 格式的遠端 IP 位址。然後，更新的工具、指令碼和軟體將正確剖析 IPv6 格式的遠端 IP 位址。

使用雙堆疊端點來透過 IPv6 網路提出請求

若要透過 IPv6 使用 S3 on Outposts API 呼叫提出請求，您可以透過 AWS CLI 或 AWS SDK 使用雙堆疊端點。無論是透過 IPv6 通訊協定或 IPv4 通訊協定存取 S3 on Outposts，Amazon S3 控制 API 操作和 S3 on Outposts API 操作的運作方式都相同。不過，請注意，IPv6 網路不支援 S3 on Outposts 物件動作 (例如 `PutObject` 或 `GetObject`)。

使用 AWS Command Line Interface (AWS CLI) 和 AWS SDKs時，您可以使用參數或旗標來變更為雙堆疊端點。您也可以直接指定雙堆疊端點來覆寫組態檔中的 S3 on Outposts 端點。

您可以使用雙堆疊端點，從下列任一位置透過 IPv6 存取 S3 on Outposts 儲存貯體：

- AWS CLI，請參閱 從 使用雙堆疊端點 AWS CLI。

- 「 AWS SDKs，請參閱「」從 AWS SDKs 使用 S3 on Outposts 雙堆疊端點。

在 IAM 原則中使用 IPv6 地址

在嘗試使用 IPv6 通訊協定存取 S3 on Outposts 儲存貯體之前，請確保用於 IP 位址篩選條件的 IAM 使用者或 S3 on Outposts 儲存貯體政策皆已更新，以包含 IPv6 位址範圍。如果未更新 IP 位址篩選政策以處理 IPv6 位址，則嘗試使用 IPv6 通訊協定時，可能會失去對 S3 on Outposts 儲存貯體的存取權。

篩選 IP 位址的 IAM 政策使用 IP 位址條件運算子。下列 S3 on Outposts 儲存貯體政策會使用 IP 位址條件運算子找出允許之 IPv4 位址的 54.240.143.* IP 範圍。拒絕此範圍外的任何 IP 位址存取 S3 on Outposts 儲存貯體 (DOC-EXAMPLE-BUCKET)。因為所有的 IPv6 地址都在允許的範圍外，所以此原則可避免 IPv6 地址得以存取 DOC-EXAMPLE-BUCKET。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3outposts:*",
      "Resource": "arn:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET/*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

您可以修改 S3 on Outposts 儲存貯體政策的 Condition 元素，允許 IPv4 (54.240.143.0/24) 與 IPv6 (2001:DB8:1234:5678::/64) 位址範圍，如下列範例所示。您可使用本例所示的同類型 Condition 區塊，更新您的 IAM 使用者與儲存貯體原則。

```
       "Condition": {
         "IpAddress": {
            "aws:SourceIp": [
              "54.240.143.0/24",
               "2001:DB8:1234:5678::/64"
             ]
          }
```

```
        }
```

使用 IPv6 之前，您必須更新所有相關的 IAM 使用者與儲存貯體原則，它們會使用 IP 地址篩選條件允許 IPv6 地址範圍。除現有的 IPv4 地址範圍外，建議您也更新 IAM 原則的貴組織 IPv6 地址範圍。如需允許透過 IPv6 與 IPv4 存取的儲存貯體原則範例，請參閱限制針對特定 IP 位址的存取。

您可以使用 https://console.aws.amazon.com/iam/的 IAM 主控台來檢閱 IAM 使用者原則。如需 IAM 的詳細資訊，請參閱《IAM 使用者指南》。如需有關編輯 S3 on Outposts 儲存貯體政策的資訊，請參閱新增或編輯 Amazon S3 on Outposts 儲存貯體的儲存貯體政策。

測試 IP 地址相容性

如果您使用的是 Linux 或 Unix 執行個體，或 macOS X 平台，您可以透過 IPv6 測試對雙堆疊端點的存取權。例如，若要透過 IPv6 測試與 Amazon S3 on Outposts 端點的連線，請使用 `dig` 命令：

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

如果 IPv6 網路上的雙堆疊端點經過正確設定，則 `dig` 命令會傳回連線的 IPv6 位址。例如：

```
dig s3-outposts.us-west-2.api.aws AAAA +short

2600:1f14:2588:4800:b3a9:1460:159f:ebce

2600:1f14:2588:4802:6df6:c1fd:ef8a:fc76

2600:1f14:2588:4801:d802:8ccf:4e04:817
```

搭配 AWS PrivateLink使用 IPv6

S3 on Outposts 支援 AWS PrivateLink 服務和端點的 IPv6 通訊協定。透過 IPv6 通訊協定的 AWS PrivateLink 支援，您可以透過 IPv6 網路，從內部部署或其他私有連線連線至 VPC 內的服務端點。AWS PrivateLink 適用於 S3 on Outposts 的 IPv6 支援也可讓您 AWS PrivateLink 與雙堆疊端點整合。如需如何為 啟用 IPv6 的步驟 AWS PrivateLink，請參閱使用 AWS PrivateLink 服務和端點加速採用 IPv6。

> **Note**
>
> 若要將支援的 IP 位址類型從 IPv4 更新為 IPv6，請參閱《AWS PrivateLink 使用者指南》中的修改支援的 IP 位址類型。

搭配 AWS PrivateLink使用 IPv6

如果您使用 AWS PrivateLink 搭配 IPv6，則必須建立 IPv6 或雙堆疊 VPC 介面端點。如需如何使用 AWS Management Console 建立 VPC 端點的一般步驟，請參閱AWS PrivateLink 《使用者指南》中的使用介面 VPC 端點存取 AWS 服務。

AWS Management Console

使用下列程序建立連線至 S3 on Outposts 的介面 VPC 端點。

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/vpc/ 開啟 VPC 主控台。
2. 在導覽窗格中選擇端點。
3. 選擇建立端點。
4. 在 Service category (服務類別) 中，選擇 AWS services。
5. 針對服務名稱，選擇 S3 on Outposts 服務 (com.amazonaws.us-east-1.s3-outposts)。
6. 對於 VPC，請選擇您將從中存取 S3 on Outposts 的 VPC。
7. 對於子網路，為每個可用區域選取一個子網路，您將從中存取 S3 on Outposts。您無法在相同的可用區域內選取多個子網路。系統會為您選取的每個子網路建立新的端點網路介面。系統預設會將子網路 IP 位址範圍的 IP 位址指派給端點網路介面。若要指定端點網路介面的 IP 位址，請選擇指定 IP 地址，並在子網路位址範圍輸入 IPv6 位址。
8. 對於 IP 位址類型，請選擇 Dualstack。將 IPv4 和 IPv6 位址指派給您的端點網路介面。只有當所有選取的子網都具有 IPv4 和 IPv6 地址範圍時，才支援此選項。
9. 對於安全群組，選擇要與 VPC 端點的端點網路界面建立關聯的安全群組。預設安全群組預設會與 VPC 建立關聯。
10. 針對政策，選擇完整存取，以允許 VPC 端點上所有資源的所有主體進行所有操作。否則，選擇自訂以連接 VPC 端點政策，該政策會控制主體在 VPC 端點上對資源執行操作所擁有的許可。只有服務支援 VPC 端點政策時，此選項才可用。如需詳細資訊，請參閱端點政策。
11. (選用) 若要新增標籤，請選擇 Add new tag (新增標籤)，然後輸入標籤的鍵和值。
12. 選擇建立端點。

Example - S3 on Outposts 儲存貯體政策

若要允許 S3 on Outposts 與您的 VPC 端點互動，您可以更新 S3 on Outposts 政策，如下所示：

```
{
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3-outposts:*",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

AWS CLI

> **Note**
>
> 若要在 VPC 端點上啟用 IPv6 網路，您必須為 S3 on Outposts 的 SupportedIpAddressType 篩選條件設定 IPv6。

下列範例使用 create-vpc-endpoint 命令來建立新的雙堆疊介面端點。

```
aws ec2 create-vpc-endpoint \
--vpc-id vpc-12345678 \
--vpc-endpoint-type Interface \
--service-name com.amazonaws.us-east-1.s3-outposts \
--subnet-id subnet-12345678 \
--security-group-id sg-12345678 \
--ip-address-type dualstack \
--dns-options "DnsRecordIpType=dualstack"
```

根據 AWS PrivateLink 服務組態，新建立的端點連線可能需要由 VPC 端點服務提供者接受，才能使用。如需詳細資訊，請參閱《AWS PrivateLink 使用者指南》中的接受和拒絕端點連線請求。

下列範例使用 modify-vpc-endpoint 命令，將僅限 IPv 的 VPC 端點更新為雙堆疊端點。雙堆疊端點允許存取 IPv4 和 IPv6 網路。

```
aws ec2 modify-vpc-endpoint \
--vpc-endpoint-id vpce-12345678 \
--add-subnet-ids subnet-12345678 \
--remove-subnet-ids subnet-12345678 \
--ip-address-type dualstack \
```

```
--dns-options "DnsRecordIpType=dualstack"
```

如需如何啟用 IPv6 網路的詳細資訊 AWS PrivateLink，請參閱使用 AWS PrivateLink 服務和端點加速採用 IPv6。

使用 S3 on Outposts 雙堆疊端點

S3 on Outposts 雙堆疊端點支援透過 IPv6 與 IPv4 對 S3 on Outposts 儲存貯體提出請求。本節說明如何使用 S3 on Outposts 雙堆疊端點。

主題

- S3 on Outposts 雙堆疊端點
- 從 使用雙堆疊端點 AWS CLI
- 從 AWS SDKs 使用 S3 on Outposts 雙堆疊端點

S3 on Outposts 雙堆疊端點

當您對雙堆疊端點時提出請求時，S3 on Outposts 儲存貯體 URL 會解析為 IPv6 或 IPv4 位址。如需如何透過 IPv6 存取 S3 on Outposts 儲存貯體的詳細資訊，請參閱透過 IPv6 向 S3 on Outposts 提出請求。

若要透過雙堆疊端點存取 S3 on Outposts 儲存貯體，請使用路徑樣式端點名稱。S3 on Outposts 僅支援區域雙堆疊端點名稱，亦即，指定的名稱必須包含區域。

針對雙堆疊端點路徑樣式 FIP 端點，請使用下列命名慣例：

```
s3-outposts-fips.region.api.aws
```

針對雙堆疊非 FIP 端點，請使用下列命名慣例：

```
s3-outposts.region.api.aws
```

> **Note**
>
> S3 on Outposts 中不支援虛擬託管樣式端點名稱。

從 使用雙堆疊端點 AWS CLI

本節提供用於向雙堆疊端點提出請求的 AWS CLI 命令範例。如需設定 的說明 AWS CLI，請參閱 開始使用適用於 Java 的 AWS CLI 和 開發套件。

您可以在 AWS Config 檔案的設定檔use_dualstack_endpointtrue中將組態值設定為 ，將 s3和 s3api AWS CLI 命令提出的所有 Amazon S3 請求導向至指定區域的雙堆疊端點。您可以在組態檔案或命令中使用 --region 選項指定區域。

搭配 使用雙堆疊端點時 AWS CLI，僅支援path定址樣式。組態檔案中設定的定址樣式會決定儲存貯體名稱應包含主機名稱中，或是包含在 URL 中。如需詳細資訊，請參閱《AWS CLI 使用者指南》中的 s3outposts。

若要透過 使用雙堆疊端點 AWS CLI，請將 --endpoint-url 參數與 http://s3.dualstack.*region*.amazonaws.com或 https://s3-outposts-fips.*region*.api.aws端點搭配使用，以用於任何 s3control或 s3outposts命令。

例如：

```
$  aws s3control list-regional-buckets --endpoint-url https://s3-
outposts.region.api.aws
```

從 AWS SDKs 使用 S3 on Outposts 雙堆疊端點

本節提供如何使用 AWS 開發套件存取雙堆疊端點的範例。

AWS SDK for Java 2.x 雙堆疊端點範例

下列範例示範如何在使用 AWS SDK for Java 2.x建立 S3 on Outposts 用戶端時，使用 S3ControlClient 和 S3OutpostsClient 類別啟用雙堆疊端點。如需建立及測試 Amazon S3 on Outposts 的可行 Java 範例的指示，請參閱開始使用適用於 Java 的 AWS CLI 和 開發套件。

Example – 在啟用雙堆疊端點的情況下建立 **S3ControlClient** 類別

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsRequest;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsResponse;
import software.amazon.awssdk.services.s3control.model.S3ControlException;
```

```
public class DualStackEndpointsExample1 {

    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");
        String accountId = "111122223333";
        String navyId = "9876543210";

        try {
            // Create an S3ControlClient with dual-stack endpoints enabled.
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                                                             .region(clientRegion)
                                                             .dualstackEnabled(true)
                                                             .build();
            ListRegionalBucketsRequest listRegionalBucketsRequest =
 ListRegionalBucketsRequest.builder()

       .accountId(accountId)

       .outpostId(navyId)

       .build();

            ListRegionalBucketsResponse listBuckets =
 s3ControlClient.listRegionalBuckets(listRegionalBucketsRequest);
            System.out.printf("ListRegionalBuckets Response: %s%n",
 listBuckets.toString());
       } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 on Outposts
 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
       }
       catch (S3ControlException e) {
            // Unknown exceptions will be thrown as an instance of this type.
            e.printStackTrace();
       } catch (SdkClientException e) {
            // Amazon S3 on Outposts couldn't be contacted for a response, or the
 client
            // couldn't parse the response from Amazon S3 on Outposts.
            e.printStackTrace();
       }
   }
```

```
}
```

Example – 建立已啟用雙堆疊端點的 **S3OutpostsClient**

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3outposts.S3OutpostsClient;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsRequest;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsResponse;
import software.amazon.awssdk.services.s3outposts.model.S3OutpostsException;


public class DualStackEndpointsExample2 {

    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");

        try {
            // Create an S3OutpostsClient with dual-stack endpoints enabled.
            S3OutpostsClient s3OutpostsClient = S3OutpostsClient.builder()
                                                                .region(clientRegion)
                                                                .dualstackEnabled(true)
                                                                .build();
            ListEndpointsRequest listEndpointsRequest =
 ListEndpointsRequest.builder().build();

            ListEndpointsResponse listEndpoints =
 s3OutpostsClient.listEndpoints(listEndpointsRequest);
            System.out.printf("ListEndpoints Response: %s%n",
 listEndpoints.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 on Outposts
 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch (S3OutpostsException e) {
            // Unknown exceptions will be thrown as an instance of this type.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 on Outposts couldn't be contacted for a response, or the
 client
```

```
            // couldn't parse the response from Amazon S3 on Outposts.
            e.printStackTrace();
        }
    }
}
```

如果您在 Windows AWS SDK for Java 2.x 上使用 ，您可能需要設定下列 Java 虛擬機器 (JVM) 屬性：

```
java.net.preferIPv6Addresses=true
```

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。